THEORY of OPERATION &
SIMPLIFIED DRAWINGS for

CONTROL DATA®
G-15 COMPUTER

Record of Revisions

REVISION	NOTES

Pub. No. 60121600
November, 1964

Address comments concerning this
manual to:

Control Data Corporation
Technical Publications Department
4201 North Lexington Avenue
St. Paul, Minnesota 55112

or use Comment Sheet in back of
this manual.

PART I - THEORY OF OPERATION

PREFACE

This manual is designed to clarify the internal logic and circuitry of the G-15. It presents both theory of operation (Part I) and simplified drawings (Part II) to illustrate the reasoning behind the circuits.

While the material presented assumes some familiarity with the G-15 as well as knowledge of Binary Arithmetic and Boolean Algebra, outlines of fundamental operations are provided in the APPENDIX, along with two fundamental test routines.

TABLE OF CONTENTS

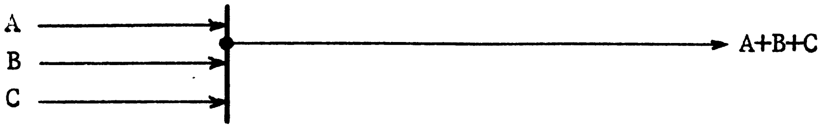
SECTION A

FUNDAMENTAL CIRCUITS

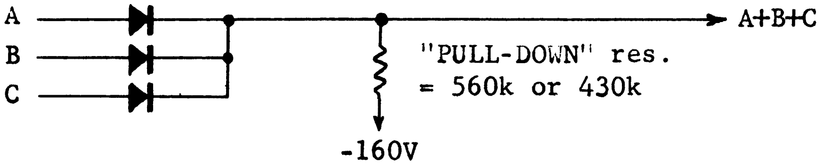
This section is devoted to the fundamental circuits which constitute the "building blocks" of which the G-15D is composed. It is suggested that the reader thoroughly digest this section prior to proceeding further, for in the sections to follow an understanding of these circuits is taken for granted.

"OR" GATES

A-1a An OR gate is a device for the purpose of yielding a high output signal if any of its input terms are high. Symbolically an OR gate is represented as shown:

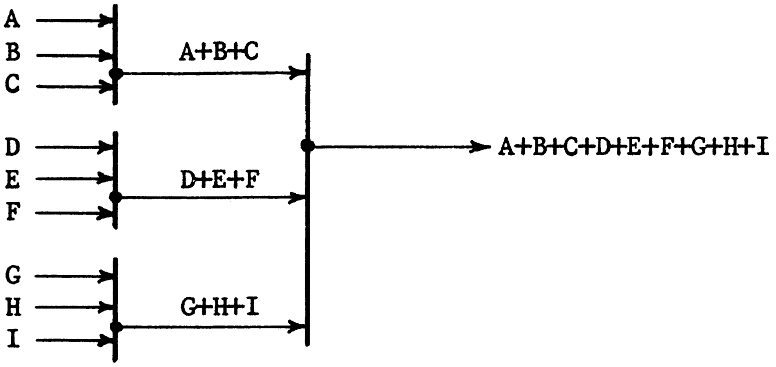


schematically:



A-1b Considering the connection of the diodes, the input terms are isolated from one another by diode back resistance. The common connection of the diodes will yield an output which, for all practical purposes, is short-circuited to the input term having the highest potential. Assuming the inputs are (individually) either at -20V or OV, the output should be at the highest input potential. That is, if one or more input terms reach OV the output will reach OV; if none are above -20V, the output will be at -20V. The loading resistance to -160V is known as the PULL-DOWN resistor; its function is to "pull" the output potential "down", while the input terms pull it up.

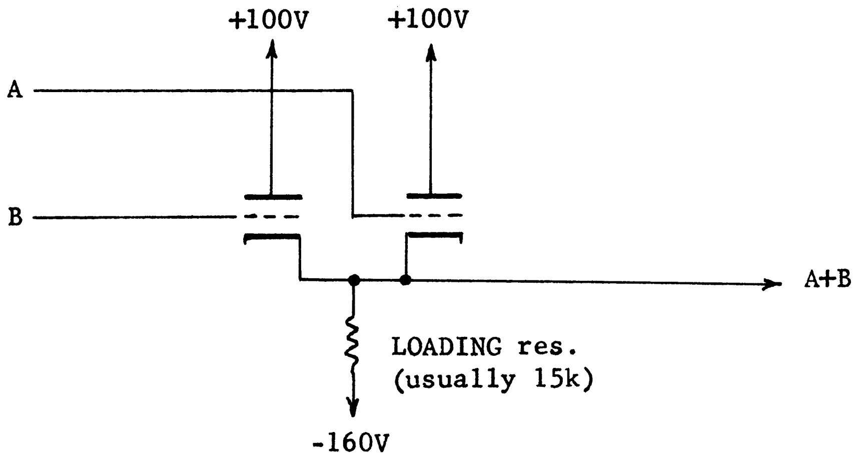
A-1c Should only one input term be high (OV), its diode must handle all of the current required to drop 160V across the PULL-DOWN resistor. The other diodes will be "biased off"; however, a certain amount of reverse current will exist in the "biased off" diodes. This will be a function of the reverse voltage applied (-20V) and the back resistance of each diode. As the number of terms entering an OR gate is increased, the back current loading problem increases. For this reason a limit is placed on how many terms can feed one OR gate. If more than the allowable limit is to be exceeded, a series of OR gates may be employed to relieve the problem. For example, if nine terms are to feed an OR gate, the following arrangement can be used:



A-1d Logically, the function is identical to that of a single nine-term OR gate. Although the total forward resistance may be increased by a trivial amount, the back resistance loading problem is substantially reduced. Drawing #3D293 illustrates the distribution of diodes comprising the "EB", which is a multi-term OR gate.

A-1e Dynamic properties: If an OR gate is fed by a square-wave, and the output is loaded by distributed capacitance to ground, the output signal will rise (from -20V to 0V) quite rapidly since the capacitance will discharge through the forward resistance of the active diode; however, when the output signal drops, the capacitance must charge through the PULL-DOWN resistor, which has a relatively high value. Hence, the decay time of the TRAILING EDGE of an OR gate's output signal is liable to be SLOW.

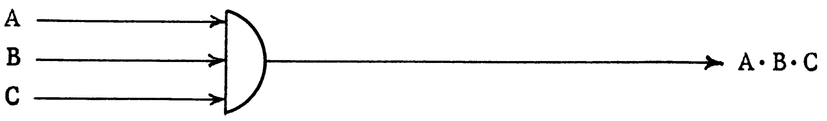
A-1f Cathode-follower OR gates: Cathode-followers may be used in the construction of an OR gate as follows:



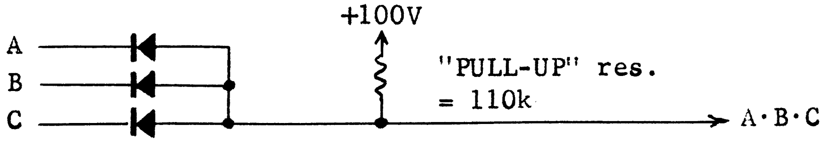
This circuit is advantageous in that (1) loading of the inputs is light, (2) inputs are isolated from the output, and (3) the circuit supplies a power boost to the output term. Only a few such gates exist in the G-15D. One such circuit may be seen on Drawing #3D293 - see "EB". (Cathode-follower OR gates are sometimes referred to as "FAST 'OR' GATES" since they do not suffer from the phenomenon described in section A-1e.)

"AND" GATES

A-2a An AND gate is a logical circuit for the purpose of yielding a high output only if all of its input terms are high. Symbolically, an AND gate is illustrated as follows:

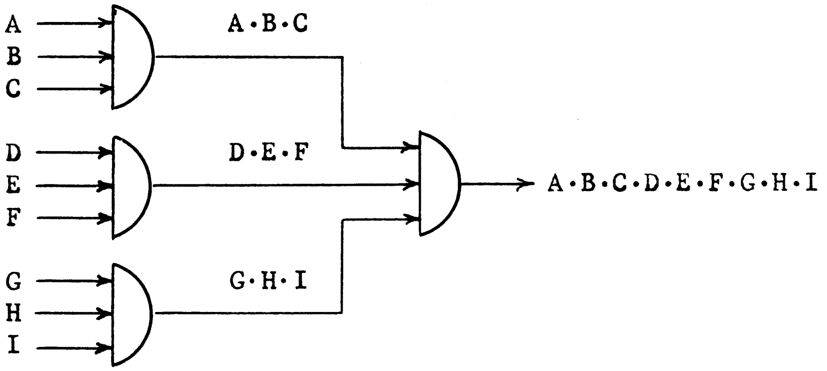


schematically:



A-2b The output is virtually short-circuited to the lowest input term. Therefore, only if all inputs are high (OV) will the output be high. The loading resistor is known as the PULL-UP resistor; its function is to "pull" the output potential "up", while the input terms pull it down.

A-2c As in the case of an OR gate, the input terms are isolated from one another by diode back resistance; also, a single AND gate with an excessive number of input terms will yield a significant back resistance loading problem. Splitting multi-term AND gates as shown below will relieve the back resistance problem (similar to the OR gate case) without altering the logic of the circuit:



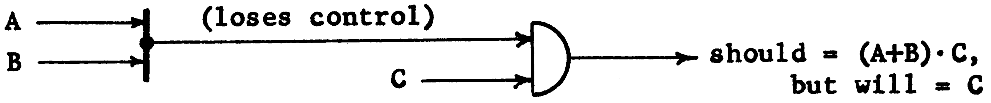
A-2d Dynamic properties: If an AND gate is fed by a square-wave, and the output is loaded by distributed capacitance to ground, the output will rise (from -20V to OV) by virtue of the capacitance discharging through the PULL-UP resistor; therefore, the rise time of the LEADING EDGE of an AND gate's output is liable to be SLOW. The trailing edge (decline from OV to -20V) will be rapid since the capacitance will charge through the forward resistance of the active diode. (In some cases, to reduce rise time, PULL-UP resistors have been shunted by "FREE RESISTORS", which are 110k and 150k resistors to +100V and are available in D1 and D2 packages.)

AND GATES and OR GATES - COUPLING

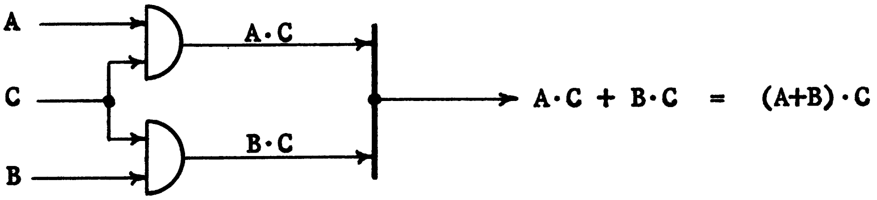
A-3a OR gates and AND gates may be connected to one another subject to the following major restriction: It is permissible for an AND gate to feed an OR gate directly; however, an OR gate may not feed an AND gate without an intervening stage of isolation. (A CATHODE-FOLLOWER or BUFFER-INVERTER can supply the necessary isolation.)

A-3b The reason for this restriction may be attributed to the fact that an OR gate (with or without PULL-DOWN resistor - ref. A-10a) is unable to carry enough current to drop 120V across the 110k PULL-UP resistor in the AND gate; consequently the OR gate cannot render the AND gate's output equal to -20V when the logic calls for such a case. Ohm's law will prove the point.

A-3c As an example, the circuit below should yield (A+B)·C, but will actually yield C since the (A+B) combination cannot disqualify the AND gate.

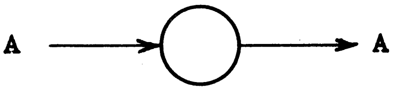


a satisfactory alternative circuit is:



CATHODE-FOLLOWERS

A-4a A CATHODE-FOLLOWER is symbolically noted as follows:



(Schematic on dwg. 13)

A-4b Cathode Followers serve no purpose as far as logic is concerned. They handle certain technicalities in the electronics. The proposition of an OR gate feeding an AND gate (above) is one example. They are installed whenever necessary for purposes such as the following:

1) Isolation (hi Z in, lo Z out).

2) Power boost
   a) Resistive loads
   b) Reactive loads (i.e. preservation of wave shape when distributed capacitance loading is heavy).

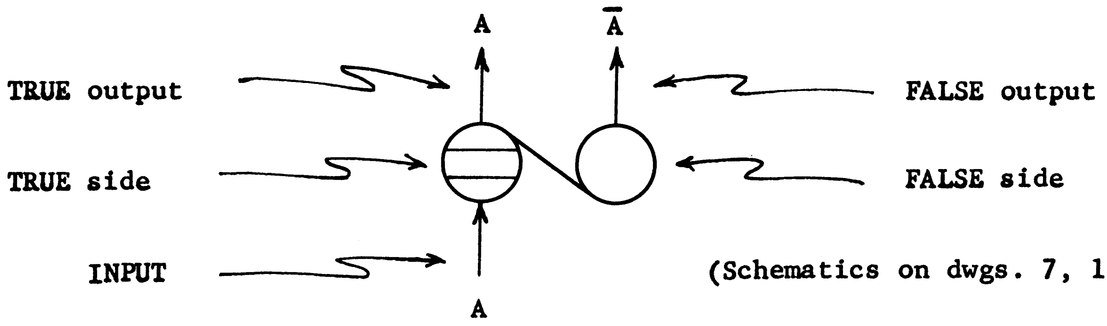
3) Addition of constant voltage to input.

A-4c The CATHODE-FOLLOWER circuits available in the packages provide loading flexibility in some cases. These cases provide optional amounts of cathode loading resistance within the CATHODE-FOLLOWER package itself, allowing a variety of loads. The grid circuits also provide a degree of flexibility in some cases.

Note that the grid return resistances in the packages involve voltage dividers. The reason for this is to render the grid potential -92V (instead of -160V), in the event of removal of the package supplying the input signal. This will prevent the cathode from approaching -160V which could result in (1) exceeding the filament-to-cathode voltage rating of the tube (filaments are returned to -55VDC), and (2) applying excessive inverse voltage to diodes controlled by the cathode-follower.

BUFFER-INVERTER

A-5a A BUFFER-INVERTER is symbolically noted as follows:



(Schematics on dwgs. 7, 13)

A-5b A BUFFER-INVERTER is a follower. Its true output "A" follows the input "A". The false output "$\bar{A}$" is the exact opposite of "A" (i.e. "$\bar{A}$" is "A" inverted, hence the term "inverter").

A-5c The outputs are clamped at -20V or 0V and are capable of feeding several loads. Furthermore, loading of the outputs does not affect the input (hence the term "buffer").

A-5d Refer to Drawing 7 (left), BUFFER-INVERTER schematic: The first inverter (V3A) receives the input signal "A" at pin 2. This signal should be at 0V, -20V, or in transition - subject to integration. The plate output should yield a potential at the grid of the next inverter (V2A) such that it will either be cut off or fully conducting. C5 and C7 both contribute to high frequency peaking to compensate for possible integration suffered by the incoming signal (V3A, pin 2) due to distributed capacitance. This peaking results in rapid transit time in the output current of V2A (and V3B), effectively "cleaning up" the leading and trailing edges of the incoming signal, hence synthesizing its ideal shape.

A-5e The V2A and V3B plate circuits are completed by components located in DIODE CLAMP package. A schematic of such a package is shown on

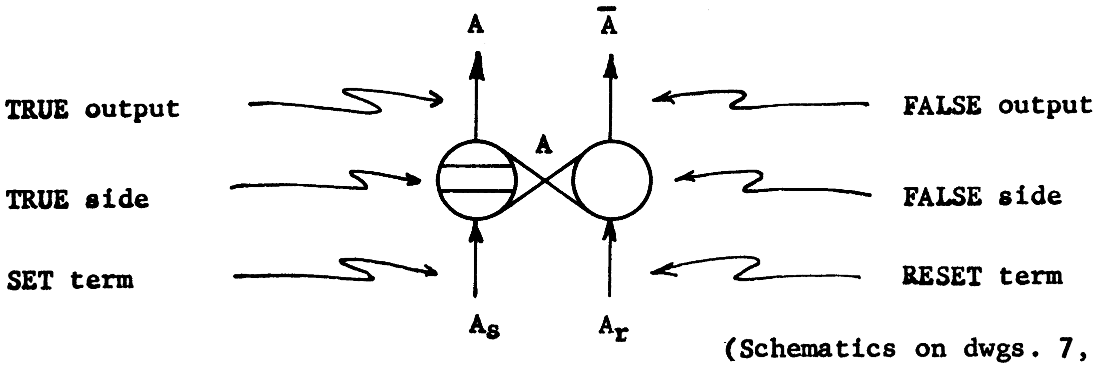
Drawing 12. This packaging system keeps the diodes physically separated from the heat-producing components in the BUFFER-INVERTER package.

A-5f When V2A conducts, its plate will be low consequently cutting off V3B; conversely, if V2A is cut off, its plate will be high causing V3B to conduct. Output pin D will yield the TRUE output "A" - pin C the FALSE output "$\overline{A}$". Normally either V2A or V3B (but not both) will conduct providing a potential of approximately -121V to their common cathode connection; C2 holds this potential during transitions.

A-5g The DIODE CLAMP circuit terminating the V2A and V3B plate circuits assure that output pins D and C will be clamped either at -20V or OV. Loading of the outputs should only serve to add to or subtract from the clamping current. Overloading of these outputs can prevent clamping current, resulting in instability. Low emission triodes can also result in inadequate current for loading and/or clamping.

FLIP-FLOP

A-6a A FLIP-FLOP is symbolically noted as follows:



(Schematics on dwgs. 7, 13)

A-6b A FLIP-FLOP is an ELECTRONIC SWITCH consisting of two D.C. inverters and a trigger circuit. It has two stable states known as SET and RESET. SET is the state in which the TRUE OUTPUT (striped side of symbol) is high (OV) while the FALSE OUTPUT is low (-20V). RESET is the state in which the FALSE OUTPUT is high (OV) and the TRUE OUTPUT is low (-20V). The SET STATE results from the application of a SET TERM, while the RESET STATE results from the application of a RESET TERM. The FLIP-FLOP once rendered in one of its two stable states (SET or RESET) will remain in that state until caused to change by an input term calling for a change in state.

A-6c Drawing 7 (right) shows a schematic of a complete FLIP-FLOP circuit. It can be seen that, with the exception of the NEON SERIES RESISTOR, the two inverters are drawn as mirror-images of each other and that each one feeds the other.

A-6d Normally, one tube will be conducting, thereby holding the other cut off. If V1A is conducting, it will hold V1B cut off maintaining the SET state of the FLIP-FLOP. V1A, by conducting, will render the FALSE output (pin T) low (-20V), while V1B, by not conducting, will allow the TRUE OUTPUT (pin S) to be high (OV). If the above state were reversed, the FLIP-FLOP would be in the RESET state.

A-6e To bring about a change in state, the conducting tube will be driven into cut-off by a negative trigger signal. This will cause a rise in its plate voltage which will turn the non-conducting tube on; this, in turn, will hold the previously conducting tube cut off. This establishes the new stable state in which the FLIP-FLOP will remain until another trigger signal, applied to the opposite side, causes it to change.

A-6f The trigger circuit consists of two identical gates: one for SET and the other for RESET. CLOCK is applied to both, CLOCK consisting of negative pulses of short duration (.3 - .5 microseconds) appearing at 9.3 microsecond intervals. The CLOCK line is normally at OV, but reaches -13V during a CLOCK pulse. CLOCK is the synchronizing signal that causes the FLIP-FLOP to be SET or RESET depending upon logical presence or absence of the SET TERM or RESET TERM.

A-6g Consider now the RESET circuit. If the RESET TERM is logically absent (at -20V), CLOCK is decoupled from C3 by virtue of the back resistance of CR1, which will be "baised off." In consequence no negative pip should be coupled through C3 to the grid of V1A. On the other hand, if the RESET TERM were to be logically present (at OV) at CLOCK pulse time, electrons would be transmitted through CR1's forward resistance into the bottom plate of C3 and to R5, which acts as a load resistance. Discharge of electrons from the top plate of C3 causes a negative-going "reset pulse" at the grid of V1A; this will cut V1A off.

A-6h Regenerative action takes over as V1A's current drops: V1A's plate voltage rises, turning on V1B. V1B's plate voltage drops, cutting off V1A, which is the triode which the "reset pulse" was cutting off in the first place. C1 and C2 accelerate the transition. The FLIP-FLOP will remain in this new state until V1B is cut off by a "set pulse."

A-6i Whenever a FLIP-FLOP experiences a change in state, the charges on C1, C2, C3, and C4 will change. Until these charges are almost stabilized, the FLIP-FLOP will not be receptive to further trigger pulses were they to occur in rapid succession. Ample time is available for this stabilization since CLOCK pulses, available for trigger, occur at 9.3 microsecond intervals (i.e. rep. rate = 108 kc.)

A-6j Drawing 7 indicates by waveform that when a CLOCK pulse is concurrent with the leading edge of the RESET TERM, the diode (CR1) will remain "baised-off." Even if the RESET TERM has a steep leading edge this will hold true. The potential at the top of the diode is the controlling potential. This potential will not rise instantly when a square-wave RESET TERM appears since C3 requires time to charge, with R5 (33k) limiting charge current. R5 and C3 then, constitute an integrating circuit in which high frequency components are dropped across R5. Hence, when CLOCK and the leading edge of a RESET TERM are concurrent, CR1 will remain "baised-off" during the first CLOCK pulse and the "reset pulse" will not appear until the next CLOCK pulse.

A-6k Should the trailing edge of a RESET TERM and CLOCK be concurrent the same reasoning holds true. As the RESET TERM drops. C3 will maintain a qualifying potential at the top of CR1 sufficiently long to permit CLOCK to be transmitted through the diode.

A-61 The RESET TERM itself does not appear to any significant extent at the grid of V1A. The high frequency components of the RESET TERM signal are dropped across R5, while the low frequency components are dropped across C3. If V1A and V1B were removed from the circuit and CLOCK were to be decoupled, the RESET TERM would appear at V1A-pin 2 as a small positive-going signal at leading-edge time and a small negative-going signal at trailing edge time. Neither of these are of any consequence unless marginal conditions exist in the circuit when operating with a weak tube. In such cases, the only malfunction would be a function of the leading-edge of a RESET TERM applied the FLIP-FLOP when it is already in the RESET state. The positive-going signal remnant at V1A-pin 2 could cause V1A to conduct; this could result in SETTING the FLIP-FLOP. This would only occur when V1B is weak and is unable to hold V1A cut off with an adequate safety margin.

A-6m Obviously the analysis of the RESET circuit applies also to the SET circuit since they are identical.

A-6n The example in the lower right section of Drawing 7 indicates FLIP-FLOP reaction to SET and RESET terms.

A-6o FLIP-FLOP SUMMARY: A FLIP-FLOP is an electronic switch which may be SET or RESET by means of CLOCK, which can be gated by the SET or RESET TERMS. Transition can take place only at CLOCK time. If a CLOCK pulse occurs at the instant a SET or RESET TERM begins to rise, it will be the following CLOCK pulse that causes transition. Also, a CLOCK pulse coincident with the beginning of the decline of a SET or RESET TERM will still consider that term high. If the SET TERM occurs during one pulse period, the TRUE OUTPUT of the FLIP-FLOP will not be high until the next pulse period. Similarly, if the RESET TERM occurs during one pulse period the FALSE OUTPUT of the FLIP-FLOP will not be high until the next pulse period.

MEMORY LINES (Ref. Drawing 9)

A-7a The recirculating memory line is the storage device used in the G-15D memory. It consists essentially of a writing station, a time delay, a reading station, and a means of regeneration of information (subject to external control).

A-7b Normal operation of a memory line is recirculation (i.e., information, once written in the line, will remain in the line unchanged and be revealed at the reading station periodically.) Drawing 9 is the schematic of Memory Line #2 which will be used as an example. The line is shown symbolically in the lower left-hand corner in the form it would be found on the "D-SIZED PRINTS."

A-7c The object of this discussion is to trace the writing of one bit of information through a recirculation cycle.

A-7d To begin, to write a ONE, a 2 microsecond WRITE PULSE is coupled to the writing pentode's control grid. This will cause flux in the write head (upper right) and will consequently influence the magnetic drum surface passing under its gap at the time. This magnetic surface approaching the write head is erased by a permanent magnet mounted on the drum shroud. The erasure represents ZERO; writing a ONE on the erased surface causes a distortion of the regular flux pattern of the surface. (Writing a ZERO involves no writing at all, hence no flux distortion.)

A-7e The recorded element is carried by the drum surface past a reading head after a delay determined by the peripheral speed of the drum and the spacing between the writing and reading heads. When the flux irregularity (resulting from writing the ONE) passes under the gap of the reading head, the reading head flux, normally constant, reverses and then returns to normal. This flux change induces an EMF resembling a sine wave in the reading head coil.

A-7f One end of the read head coil is grounded such that the floating end yields a sine wave, the trailing peak of which is positive. Resistance loading of the coil is accomplished in the PREAMPLIFIER CHASSIS and may be varied from 2.2k to 12.2k by a screwdriver adjustment. This is adjusted to minimize ring effect and to shunt the signal to obtain a normal level (500 mv. approx.). This is a factory adjustment and should not require attention unless the associated head is replaced.

A-7g At this point, the signal requires amplification, a matter which is accomplished in a convential plate-loaded, RC-coupled amplifier employing low loading resistance for good high frequency response. The output is cathode-follower driven out of the PREAMPLIFIER CHASSIS into a READ AMPLIFIER package where monitoring of the signal takes place.

A-7h The first step in the READ AMPLIFIER is to amplify, rectify, then amplify the incoming sine wave to produce a half sine wave of sizable swing for each detected ONE.

A-7i The first operation in amplification is to terminate the cathode follower with a step up (1:4) transformer. (The .27 mfd and 120 ohm combination in the primary circuit is to prevent peak clipping without introducing degeneration and can be considered a biasing requirement.)

A-7j The secondary of the step-up transformer constitutes the input to a "bootstrap" amplifier, which in this case is an over-biased cathode-loaded amplifier. The purpose of the amplifier is to amplify the positive portion of the input wave, without phase inversion. Response is controlled by returning the transformer secondary to a potentiometer in the cathode circuit. This adjustment controls: (1) the amount of degeneration, hence gain, and (2) the static bias on the stage, hence sampling level. It is normally adjusted to yield signals reaching the -50v level at test point K. This potentiometer may occasionally require adjustment to compensate for changing parameters elsewhere in the read-write circuit.

A-7k At this point, the half sine waves are probed by READ CLOCK which is applied to the cathode of a flip-flop trigger tube. In the event of coincidence between READ CLOCK and a ONE waveform, the trigger triode will conduct, setting the READING FLIP-FLOP to ONE. When the trigger triode conducts, grid current is inevitable; this is limited by a 2.2k grid resistor. The 15 mmf. grid-to-ground capacitance counteracts the degenerative effect on the positive going grid signal by the negative going cathode and plate signals. READ CLOCK is obliged to supply grid and plate current to all trigger tubes. This accounts for relatively high peak power requirements of the READ CLOCK pulse driver (Ref. B-3g). Since this grid current load will only present itself when a line yields a ONE, the total load will be a function of the memory contents.

A-7l Approximately 8.3 microseconds after the ONE bit has been stored in the READING FLIP-FLOP, CLOCK unconditionally resets it to ZERO. The flip-flop's ONE output is coupled via gates in the DIODE 3 (D3) package to the WRITE AMP package if the line is recirculating. (Heavy lines in the DIODE 3 package show recirculation path.)

A-7m The first item in the WRITE AMP is a cathode-follower which permits the output of the OR gate in the DIODE 3 package to feed the AND gate feeding a power pentode (Ref. A-3, A-4).

A-7n For the last two microseconds prior to resetting the READING FLIP-FLOP to ZERO, the AND gate in the WRITE AMPLIFIER was qualified by WRITE PULSE. This causes the writing pentode to conduct for two microseconds through the write head, rewriting the original ONE. Had the read head "detected" a ZERO (absence of a ONE), no writing would have taken place, effectively recirculating the ZERO.

A-7o Needless to say, prior to the leading edge of the WRITE PULSE, the ONE output of the READING FLIP-FLOP should have risen to OV. This allows 6.3 microseconds for rise time. (This 6.3 microsecond rise time requirement factor applies to most signals in the computer and accounts for many CATHODE-FOLLOWERS and resistance-loading of certain circuits - such as AND gates. Decay time is equally important but usually not as critical as rise time.)

A-7p It can be seen that the READING FLIP-FLOP activity depends upon CLOCK for reset and READ CLOCK for set. These pulses synchronize all memory lines with a standard timing reference. WRITE PULSE synchronizes the writing. These pulses are shown on the top of Drawing 6 in approximate terms; their formation is described in Sec. B-3.

A-7q The spacing between read and write heads is governed by the capacity of the particular memory line. In the case of a long memory line, the delay in the line is such that from the time a ONE is written until the time it is rewritten, 3132 (i.e., 108 x 29) pulse periods will elapse. This allows 3132 different bits to be read and rewritten within a recirculation cycle.

A-7r The short lines have shorter delays as follows: 116 pulse periods, 58 pulse periods, and 29 pulse periods. The spacing from read head to write head is known as the TANGENTIAL adjustment. It is made at the factory and should not require any further adjustment. As long as READ CLOCK probes the reading wave at its peak (test pt. K), the adjustment is satisfactory. Misadjustment would be revealed in misphasing of these two signals. CUSTOMER ENGINEERING should be consulted for adjustment details. It is unwise to attempt head adjustment without being well informed as to procedure. Accidental damage to a track could require replacement of the drum.

A-7s Another adjustment of the heads is RADIAL adjustment. This should render head (gap) to drum surface spacing equal to 1 mil. (for both read and write heads.) Proper spacing is essential for good resolution and signal-to-noise ratio. This too is a factory adjustment. Only in the case of subnormal reading waves (not attributable to failing electronic components) should RADIAL adjustment be considered. As in the case of TANGENTIAL adjustment, CUSTOMER ENGINEERING should be consulted.

A-7t Noise mentioned above is insignificant compared to the amplitude of a legitimate ONE being read since it is normally well below the sampling level in the READING FLIP-FLOP trigger stage. The worst noise element will result from crosstalk (magnetic coupling) between the write and read heads of a line. This interference is at its worst in the short (29 bit) memory lines.

A-7u Another factor which may cause deviation from optimal memory line functioning is the result of head misadjustment due to thermal effects. Therefore, if the drum is cold, a line may operate with reduced safety margins. Only a line already in need of maintenance would fail due to this effect. If any memory line adjustments are to be made, it should be done when the drum has reached its normal operating temperature.

MEMORY LINES: READ-OUT, READ-IN FUNCTIONS

A-7v READ-OUT: When information is to be read out of a line, the READ FLIP-FLOP output is coupled to an external circuit by an AND gate qualified by the desired terms. In the case of MEMORY LINE 2 (Drawing 9, lower left), M2 reaches EB if qualifying terms SW and SO are both high. Read-out does not disturb the contents of the source line.

A-7w READ-IN: When information is to be read into a line from an external source, two operations are necessary: (1) allow the new information to reach the writing circuit, and (2) simultaneously block the recirculation path of the old information.

A-7x Item (1) is handled by qualifying an AND gate to which the new information is applied. In LINE 2 (Drawing 9) terms DO and DW permit LB to be written. Item (2) is handled by blocking the gates which allow the READ FLIP-FLOP output to reach the writing station. If LINE 2 is selected as destination, both $\overline{DO}$ and $\overline{TR\cdot DW}$ will be low at the appropriate time.

A-7y Some special purpose memory lines may deviate from this pattern and will be discussed later. Needless to say, all READ-IN - READ-OUT activities are appropriately synchronized by control circuits.

THE MEMORY

A-8a The DRUM MEMORY is composed of many recirculating memory lines such as the one just described. These lines can differ from one another as follows:

1) Capacity
2) Writing control
   a) Recirculation gating
   b) Method of writing new information

A-8b The MEMORY consists of:

21 long lines - 3132 bits (108 words) - Lines 00-19, and CN
5 short lines - 116 bits (4 words) - Lines 20-23, and MZ
3 short lines - 58 bits (2 words) - ID, MQ, PN
2 short lines - 29 bits (1 word) - AR, CM

A-8c Three of the above are not directly available to the programmer since they are for the benefit of internal computer functions. These are CN, MZ and CM. In addition to the recirculating lines mentioned above there are three permanently recorded (non-recirculating) lines known as CLOCK TRACK SPARE CLOCK TRACK, and TIMING TRACK (TM). These are for timing purposes and are discussed in Sections B-3a and B-5a.

A-8d Physically, the short memory lines are sandwiched in between the read heads and erase magnet of the long lines as illustrated on Drawing 8.

A-8e Memory line specifications:

pulse period = 9.3 microseconds
packing density = 90 bits/inch
read-record mode - RETURN TO ZERO
maximum access time = 28.5 milliseconds

PHYSICAL LOCATION OF LOGICAL CIRCUITS

A-9a The fundamental circuits mentioned in the preceding sections (with the exception of READ PREAMPS) are physically located in standard packages distributed about the logic panels of the computer and are interconnected by appropriate inter-package wiring. The package schematics may be found on Drawings 11, 12 and 13. Drawing 10 provides an index of information pertaining to all packages.

A-9b The detailed logical block diagrams ("D-SIZED PRINTS") provide information relating to package locations and package jack pin-connections. Drawing 9 (MEMORY LINE 2) provides an example of a detailed logical block (lower left) and the schematic components and packages it represents. Package locations are designated by coordinates, a letter and a number, which are printed on the jack side of the logic panels themselves.

A-9c Common sense should render the notations self-explanatory with the exception perhaps of CLAMP PACKAGE locations. CLAMP PACKAGES are not shown on any block diagrams, but a notation external to the associated FLIP-FLOP or BUFFER-INVERTER supplies the location information. (On Drawing 9, lower left, below FLIP-FLOP "M2", see notation "D E22 C". This states that the CLAMP PACKAGE is located in package jack with coordinates E-22 and that pin connections D and C are used.)

A-9d A few self-imposed exercises requiring translation of information contained in the detailed block diagrams to schematic form should reveal the scheme to the reader.

A-9e Technical details of circuits not included in standard packages will be discussed as they arise.

A-10 MISCELLANEOUS

a) PULL-DOWN resistors associated with OR gates are physically located in the package containing the load presented to the OR gate (e.g. Drawing - 9 - input to WRITE AMP).

b) PULL-UP resistors associated with AND gates are located (1) within the package where the AND gate diodes are, or (2) another package, or (3) in more than one package (when paralleled). Reducing PULL-UP resistance by parallel combinations can be for the purpose of compensating for low PULL-DOWN resistance in a network of gates and/or to reduce rise time of a signal.

c) The diodes associated with a single gate are not necessarily all located in the same package. Pre-wired combinations within Dl and D2 packages provide a variety of options minimizing wiring external to the package.

d) The secondary of the 6.3V filament transformer which feeds all filaments is returned to -55V to minimize possibility of exceeding the heater-to-cathode voltage ratings of all tubes.

e) Utilization of a right-to-left time axis is used as a convenience in analyzing most circuit activity (ref. G-10b).

SECTION B

BASIC TIMING

This section is devoted to the basic timing subdivisions and associated signals involved in the G-15D. The reader is urged to memorize the nature of the timing structure and signals before proceeding to sections to follow since Boolean algebra notations define times merely in terms of signals. It is not necessary to digest all of the details of the electronic circuits which give rise to the various timing signals before reading the sections to follow.

G-15D IN GENERAL

B-1a The G-15D is composed of a combination of logical "building blocks" discussed in Section A. The heart of the computer may be considered the magnetic drum which not only constitutes the MEMORY, but also the source of all timing signals.

B-1b All computer activities are synchronized with the basic timing signals mentioned above in a consistent orderly manner. Since such timing signals control logical activity, the time axis may be considered a standard reference in describing an event, for this reason timing warrants discussion prior to any other functions.

TIMING SUBDIVISIONS

B-2a (Ref. Drawing 6) In general, time in the computer is cyclic as it is on a clock. The basic cycle is known as a DRUM CYCLE (sometimes called a LOGICAL DRUM REVOLUTION), during which the entire contents of the memory are revealed and can be subjected to alteration. A DRUM CYCLE is the recirculation time of a long (3132 bit) memory line and is arbitrarily divided into 108 WORD TIMES (WT) (as a day, which is a cycle, is divided into 24 hours.) These WORD TIMES are numbered 00 - 107.

B-2b Each WORD TIME is divided into 29 PULSE PERIODS (PP) designated T1 - T29.

B-2c Associated with each PULSE PERIOD is a trio of pulses, the most significant one of which is known as CLOCK. CLOCK pulses occur at the junction of two adjacent PULSE PERIODS. The leading edges of these CLOCK pulses cause the transitions in most flip-flops. A CLOCK pulse, the duration of which is logically insignificant (.3 - .5 microseconds), may be considered as designating either the beginning of one PULSE PERIOD or the end of the preceding one. The remaining two pulses of the trio are READ CLOCK and WRITE PULSE, and are necessary for the operation of memory lines. (Their use is restricted to internal functions of the memory lines only, hence they need not be considered as constituting significant general purpose logic pulses.)

B-2d Drawing 6 illustrates the sequence of the above mentioned elements of time. Below is a convenient analogy between the G-15D computer time reference and the familiar times of day:

1 DRUM CYCLE	-----1 day
1 WORD TIME	------1 hour
1 PULSE PERIOD	-------1 minute

B-2e Just as events that occur during any day may be described in terms of hours and minutes, a computer event may be designated by word time and pulse period (e.g.,T29 of Word 107 is the last PULSE PERIOD of a DRUM CYCLE). Also, just as a duration of activity may be described as lasting for N hours, a computer activity may be said to last for N WORD TIMES.

B-2f Unless the computer is caused to alter its internally stored information, any subdivision of information will be available every DRUM CYCLE at the same time in terms of WORD TIME(s) and PULSE PERIOD(s).

B-2g The basic element of information circulating in the computer is the BINARY DIGIT (abbrev: BIT). Timewise, a particular bit of information may be located by designating its PULSE PERIOD and WORD TIME.

B-2h The basic information group consists of 29 bits and will "occupy" one WORD TIME (1 WT = 29 PP). Such a group is known as a WORD and may represent anything the programmer may designate (i. e., an instruction, a number and sign, part of a number, a control code, etc.). It is up to the programmer to determine the information content of the WORDS and their disposition. The computer itself will only do what it is instructed to do.

B-2i In order to logically and electronically identify certain times, it is necessary to have available certain signals which will occur periodically at the designated times. For this purpose three tracks (plus a spare) are available on the drum. These tracks, in conjunction with circuits known as CLOCK CHASSIS, TIMING GATES and CONTROL GATES, yield the necessary signals. Drawing 15 indicates the system.

WRITE PULSE, CLOCK, READ CLOCK - UNCONDITIONAL

B-3a CLOCK TRACK: This trio of signals occurs unconditionally every pulse period.Their origin is the CLOCK TRACK (see Drawing 15 - lower left). This track is permanently recorded around the drum (i.e., it has only a reading head and hence does not recirculate). The reading wave from the head resembles a continuous sequence of sine waves. The manner in which this track is recorded is such that exactly 3596 (i.e., 124 x 29) pulses are recorded around the track in a closed circle. This recording procedure is performed at the factory and would constitute a difficult problem to re-record in the field should an accidental erasure occur. (A grid to plate short in the CLOCK TRACK preamplifier tube could cause track erasure by DC current in the read head.) For this reason a spare recorded CLOCK TRACK is provided. Should the active CLOCK TRACK become erased, the spare may be used until the normally active track is re-recorded. In the unlikely event that re-recording of an erased CLOCK TRACK is necessary, instructions for doing so in the field can be obtained from CUSTOMER ENGINEERING.

B-3b CLOCK CHASSIS: The CLOCK TRACK reading head output is amplified by a pre-amplifier circuit identical to those used with other heads (Drawing 11). The preamplifier output is connected to the input of the CLOCK CHASSIS, which is located in the bottom of the computer (front center). The CLOCK CHASSIS (Drawings 14, 3D293) forms four different output pulses, three of which constitute the trio mentioned above.

B-3c WRITE PULSE: The preamplifier output (a series of sine waves) is "squared" by V1 and V2 in the CLOCK CHASSIS (Dwg. 14). The V2B output, differentiated, triggers multivibrator V3, which yields a 2 microsecond positive-going square wave at V3, pin 1. This 2 microsecond square wave causes power pentode V7 to conduct through its plate transformer T3. The negative end of T3's secondary (term. 4) is returned to -20V; the positive end (WRITE PULSE) is appropriately terminated to -20V and clamped at 0V. Secondary termination is accomplished by (1) a 680 ohm resistor (R29) (2) the load, and (3) two CLOCK CLAMP PACKAGES - one in each logic panel (see also dwg. 12). (Termination in the logic panels reduces transient problems.)

B-3d CLOCK: The trailing edge of the V3 multivibrator output (pin 6) causes blocking oscillator V4 to cycle yielding a positive-going pulse of .3-.5 microsecond duration at T6, term. 5. This signal, which exists directly after the decline of the WRITE PULSE, causes power pentode V5 to conduct through plate transformer T1. The positive end of the secondary (term .3) is returned to ground; the negative end (CLOCK) is appropriately terminated to ground and clamped at -13V. Secondary termination, as in the case of WRITE PULSE is accomplished (1) in the CLOCK CHASSIS (a 100 ohm res-R28, and CR3), (2) by the load, and (3) in the CLAMP package.

B-3e READ CLOCK: These pulses, used in the READ AMPLIFIER circuits to probe for the presence of ONES, are required to follow CLOCK pulses (unconditionally) by approximately 1 microsecond. They resemble CLOCK in shape, but have a base line of approximately -25V and reach a negative peak of -75V (clamping level). These two levels are both adjustable and may be varied for optimal results.

B-3f To produce READ CLOCK, CLOCK is coupled to the grids of V8 and V9 (power pentodes) via T4 (for inversion and -20V return) and via a 1 microsecond delay line. The result is that the power pentodes conduct through the plate transformer T5 for .3-.5 microsecond at the appropriate time.

B-3g Due to the nature of the load (ref. Section A-7k) and the amplitude of the READ CLOCK pulse itself, four pentode drivers are required to supply the necessary power output. V8 and V9, in the CLOCK CHASSIS itself, are supplemented by two more power pentodes in the READ CLOCK package (the only one of its kind in the computer - location: J-54). The READ CLOCK package contains two power pentodes in parallel feeding one plate transformer. The pentode grids receive the same signal as those in the CLOCK CHASSIS (TS2-9); the transformer secondary is wired in parallel with the READ CLOCK output of the CLOCK CHASSIS (i.e., the T5 secondary).

B-3h The transformers' secondary termination is accomplished in the same general manner as termination of the other signals; however, -25V and -75V are required for reference level and clamping level. V10 and V11 supply these voltages as follows:

B-3i The cathode of V11 yields a potential which is primarily a function of the setting of potentiometer R54. This cathode potential is adjusted to approximately -75V (by R54) and should not vary significantly as a result of loading. V10's function is similar to that of V11 except that V10's cathode will be primarily a function of the setting of potentiometer R51. R51 is set to yield approximately -25V from V10's cathode.

B-3j The reader will note that the -75V output is coupled to the V10 grid circuit and that an inter-action will exist. Furthermore, neon NE2-1 couples the entire amount of any V11 cathode voltage fluctuation to V10's grid, hence will tend to vary the -25V output level. This interaction tends to neutralize the effects of variations in clamping current required by READ CLOCK. (The amount of clamping current is a function of the total number of ONES in the memory.) In adjusting R51 and R54, a "rocking-in" procedure should be followed to allow for this inter-action. Although the two output potentials are called "-25V" and "-75", they are, in practice, adjusted to any levels in the vicinity which yield the most stable reading amplifier behavior.

B-3k A summary of these three pulses may be seen at the top of Drawing 6.

SHIFT PULSE - CONDITIONAL

B-4 (This signal is not among the basic pulses appropriate to mention at this point; however, its formation may as well be discussed while CLOCK CHASSIS is being described.) T6, term.5, which yields CLOCK, also feeds the grid circuit of power pentode V6. However, if the SHIFT COMMAND INPUT (= RC·CJ), which is applied to CR2, is at -20V (one stable state), the grid will not be able to rise above -20V due to clamping by CR2 - hence no V6 output. If the CR2 cathode is at OV (rather than -20V) V6 will conduct and the T2 secondary will yield a signal of the same type as CLOCK. Clamping follows the same pattern as that for CLOCK except that termination only exists in one of the two logic panels since the entire load is located in that panel only. In general, SHIFT PULSE constitutes a burst of CLOCK pulses dependent upon an externally generated signal = RC·CJ. (Sec. C-5b.)

TIMING PULSES - THE TIMING TRACK (TM)

B-5a T1, T2, T13, T21, T28 and T29: These pulses are required to identify their respective pulse periods every word time. The TIMING TRACK (TM), a permanently recorded track, supplies the necessary information to signify these pulses. The TIMING TRACK, like the CLOCK TRACK, is recorded at the factory. It repeats the same digital configuration 124 times around the periphery of the drum in a closed loop. (It is not necessary for this track to recirculate to stay in step with a DRUM CYCLE since the digital configuration in all WORD TIMES is identical and the only requirement is that an integral number of word times be recorded around the drum.)

B-5b The subdivision of each WORD TIME is illustrated on Drawing 6 (see TIMING TRACK.) The hardware is indicated on Drawing 15. The output of the TIMING TRACK READING FLIP-FLOP (TM) feeds an array of flip-flops and gates in the TIMING GATES. The key outputs from the circuit are high during T1, T2, T13, T21, T28 and T29. Generation of these signals from TM is illustrated on the timing chart on Drawing 15 - lower left. The circuit is self-synchronizing.

B-5c It is not necessary to have a spare TIMING TRACK as was the case with the CLOCK TRACK. A new TIMING TRACK may be recorded in the field if an accidental erasure occurs. If the case arises, the procedure can be obtained from CUSTOMER ENGINEERING.

WORD TIME INFORMATION - THE NUMBER TRACK (CN)

B-6a The NUMBER TRACK is the word time reference track and supplies the computer with several pieces of information. Drawing 6 (right) illustrates the contents of all words. The purpose of all bits of information will be discussed in this manual as the need for them arises. At this point only one bit need be mentioned: the bit stored at T29 time during each word time. This bit is a ONE during all word times other than WT 107, but during WT 107 itself the bit is a ZERO. The latter may be defined in Boolean algebra terminology as T29·CN, where CN is the READING FLIP-FLOP of the NUMBER TRACK.

This particular signal comes high during the last PULSE PERIOD of a DRUM CYCLE and constitutes an index or ORIGIN PULSE. The signal is called TO (i.e., TO = $T29\cdot\overline{CN}$), and is used for many purposes. Associated hardware may be found on Drawing 15.

B-6b Needless to say, since an actual physical drum revolution and a DRUM CYCLE differ (by a factor = 124/108), it is necessary that the NUMBER TRACK be a recirculating memory line so its contents will "keep step" with DRUM CYCLES. Since power turn-off involves erasure of all recirculating memory lines, a scheme of renewing the NUMBER TRACK during the power turn-on cycle is provided. This is discussed in Section G.

B-6c Once TO is established, a circuit in the CONTROL GATES (Drawing 15) involving the CE and CF flip-flops yields more timing signals. The CE flip-flop (synchronized by the NUMBER TRACK) yields an output "CE" which is high during EVEN word times only. (Obviously "$\overline{CE}$" will be high during ODD word times.) CF will, under control of CE, etc., be high during word times congruent to 2 and 3 modulo 4.

B-6d CE and CF, once synchronized, will allow the formation of certain other basic timing pulses in the TIMING GATES as follows:

TE = $T1\cdot CE$ (i.e., T1 of EVEN word times)
TF = $T29\cdot\overline{CE}\cdot CF$ (i.e., T29 of words congruent to 3 mod. 4)

B-6e The reader will note that the signal TE, which has been defined as $T1\cdot CE$ is actually formed by $T29\cdot\overline{CE}$ delayed one pulse period. Frequently the means of forming a signal is not the most straight-forward one. This can usually be attributed to conservation of hardware or preservation of waveshape. Ordinarily a signal is defined by Boolean algebra notation in terms of the conditions under which it occurs -- such notation usually follows the method whereby the signal is formed, but not necessarily.

B-6f Drawings 15 (upper left) and 6 (left) illustrate the signals just encountered. The reasons for them will become apparent later. At this point the reader should attempt to memorize the contents of Drawing 6 with the exception of details of the NUMBER TRACK not discussed in this section. These basic timing signals represent the time references for most computer functions to be discussed in sections to follow.

SECTION C

GENERAL COMPUTER ACTIVITY

This section is devoted to a variety of concepts and circuits. The circuits covered are controlled to some extent by the SPECIAL COMMANDS and other functions to be covered in sections to follow. The sequence of presentation in this particular section is less orderly than that to be found in others. This is necessary to present pre-requisite material first and to minimize forward references.

COMPUTER SEQUENCE OF ACTIVITY

C-1a Unless the computer is idle, it is engaged in a computing cycle illustrated on Drawing 18. The productive portion of the cycle takes place during the time the computer is in the TRANSFER (TR) state. It is during this time (an integral number of WORD TIMES) that an operation is executed. The TRANSFER state is sometimes referred to as the EXECUTE state.

C-1b Needless to say, the computer must know the nature of the operation to be executed; this is determined during the READ COMMAND (RC) state. During this state (lasting one WORD TIME), a command is "read" from a predetermined word location of a pre-determined line (ref. Sec. D-6b). The instant the final bit of the command (T29·RC) has been read, the command has been decoded and has set up all the necessary electronic paths for proper execution of the operation designated by the command.

C-1c Once a command has been read (and decoded) it may be executed immediately provided the command itself calls for immediate execution. If the command is not an IMMEDIATE command it is DEFERRED. As Drawing 18 illustrates, if a command is IMMEDIATE , as soon as the command is read, the TRANSFER state is initiated. If DEFERRED, the computer will rest in the WAIT TO TRANSFER (WTR) state for an integral number of word times until information in the command itself calls for the TRANSFER state.

C-1d How long the computer will remain in the TRANSFER state is also a function of the command.

C-1e Once the TRANSFER state is terminated, the computer will advance to the READ COMMAND state immediately or after an integral number of word times in the WAIT TO READ COMMAND (WRC) state. At this point the cycle repeats unless the computer is purposely caused to stop; in this case it will idle in the WAIT TO READ COMMAND state indefinitely.

COMMANDS

C-2a A command, as mentioned above, controls the computer in regard to timing (i.e., advancement from state to state), and in regard to the nature of the operation to be performed during the designated time of TRANSFER. A command is considered subdivided into two general portions: the STATIC PORTION and the DYNAMIC PORTION. (These names not only describe their purposes, but also their treatment within the computer.) Breakdown of a command is illustrated on Drawing 5 (top.)

THE STATIC PORTION: 13 bits (T1 - T13)

C-2b SOURCE AND DESTINATION (S & D): This information is contained in 10 bits (T2 - T11) which can assume 1024 (i.e. 2^{10}) different configurations, hence there are 1024 different SOURCE - DESTINATION combinations possible. Individually the SOURCE and DESTINATION portions can express SOURCES and DESTINATIONS numbered 00 - 31 (i.e., 00000 - 11111 in binary). If the DESTINATION is specified 00 - 30, the command calls for a transfer of information from a SOURCE (00-31) to the designated destination. However, if DESTINATION is selected equal to 31, the SOURCE number (00 - 31) specifies a SPECIAL COMMAND, not necessarily involving transfer of information at all.

C-2c CHARACTERISTIC (CH): This information is contained in 2 bits (T12 and T13) and can assume 4 (i.e., 2^2) different configurations, designating four options in performing a command defined by S and D. The manner in which the CH code is interpreted (hence, its result), is a function of the particular S and D selected.

C-2d SINGLE OR DOUBLE PRECISION (S/D) bit: This bit (T1) is the only remaining bit of the STATIC PORTION of a command. Its functions depend to some extent upon the remainder of the command (including the DYNAMIC PORTION). It is primarily concerned with setting up logical circuits to comply with timing requirements of SINGLE and DOUBLE precision operations; it does, however, serve other purposes.

THE DYNAMIC PORTION: 16 bits (T14 - T29)

C-2e IMMEDIATE or DEFERRED (I/D) bit: This bit (T29) controls whether the TRANSFER state will be initiated directly after the command is read or wait (in the WAIT TO TRANSFER state) until some other factor calls for it. A "0" specifies IMMEDIATE; a "1", DEFERRED.

C-2f TIMING NUMBER (T): This information is contained in 7 bits (T22 - T28) and can express a number from 00 - 127. This is a controlling factor in either initiating or terminating TRANSFER. The manner in which it is interpreted depends upon other factors in the command.

C-2g NEXT COMMAND LOCATION (N): This information is contained in 7 bits (T14 - T20) and is used to express a number from 00 - 107. These numbers refer to the WORD TIME during which the command to follow is to be read after the current command has been concluded.

C-2h BREAK POINT (BP) bit: This bit (T21) provides a programmable method of interrupting operation of the computer after the command has been executed. The setting of the COMPUTE switch on the BP position will cause the computer to idle in the WAIT TO READ COMMAND state after the TRANSFER state has been terminated.

C-2i (LOCATION of command) (L): This is not actually written as part of the command but is a piece of information associated with it. L is the word time (00 - 107) during which the command is read and usually is N of the previous command.

COMMANDS-SUMMARY

C-2j Commands consist of several subdivisions, all of which contribute to define and control the operation to be performed. The programmer is required to account for all 29 bits of a command; the engineer should be able to determine exactly what will happen within the computer as the result of any command whether or not its composition is legitimate.

NUMBERS

C-3a Ordinarily, the words to which commands make reference contain numbers or portions of numbers. Any word (in the memory) to which a program never makes reference can contain anything--usually all ZEROS.

Numbers are of two general types: SINGLE PRECISION and DOUBLE PRECISION.

C-3b SINGLE PRECISION numbers are held entirely within a single word and consist of a total of 29 bits as follows: 28 (T2 - T29) to express the number and 1 bit (Tl) to hold the sign (SIGN BIT = "0" means +; "1" means -). A single precision number can, by virtue of 28 bits, accommodate 2^{28} different binary configurations which amounts to precision up to one part in 268,435,455 (i.e. 1 part in 2^{28}-1).

C-3c DOUBLE PRECISION numbers require two words and consist of an EVEN numbered word (time) for the lowest order portion and the following ODD numbered word for the highest order portion. Sign is held in the Tl position of the EVEN word (TE = Tl$\cdot$CE), and the remaining 57 bits are devoted to expressing the number. The numerical portion can assume 2^{57} different configurations amounting to precision of one part in 144,115,188, 075,855.871 (i.e. 1 part in 2^{57}-1).

C-3d When an operation dealing with SINGLE PRECISION numbers is executed, Tl pulses constitute the TS or SIGN TIME pulse; DOUBLE PRECISION operations require TS to consist of TE (i.e., Tl$\cdot_{CE}$). This is one function of the S/D precision bit of a command (ref. drawing 15-upper right).

C-3e Whenever a number is transferred from a SOURCE to a DESTINATION, and is subject to modification in the INVERTING GATES, the modification is a function of the SIGN associated with each individual number being transferred. Obviously pulses interrogating the signs of numbers must occur at the appropriate time. Proper programming of the S/D precision bit of a command therefore is essential.

COMPLEMENTATION OF NUMBERS

C-3f Numbers are normally to be found (in memory) in terms of ABSOLUTE VALUE and SIGN. Furthermore, the BINARY POINT may be assumed to be to the left of the highest order bit of each number, rendering the highest order bit in the 2^{-1} column. (A programmed scaling factor can render such a fractional number a multiplier for 2 raised to any power, hence, order of magnitude is unrestricted - this is the programmer's concern.)

C-3g Should a number be negative, it is desirable (in order to facilitate certain operations such as addition) to convert it to COMPLEMENT form, such that given a fractional number = .N, its COMPLEMENT is (1-.N). The COMPLEMENT represents a number, which if increased by a positive value =.N, will yield all ZEROS in its fractional part and an END CARRY into the imaginary 2^{o} position. Such an END CARRY, if added to the imaginary "1" in the 2^{o} position, would yield a sum bit of "0" and an end carry which will be killed.

C-3h Evidence of the imaginary "1" in the 2^{o} position is the content of the SIGN BIT. (i.e., a "1" in the sign bit represents a negative number; if the number is in complement form, the "1" (from "1-.N") is held in the SIGN BIT.) Ideally the SIGN BIT, which has a quasi-numerical significance, should appear in a bit location following the highest order numerical bit (2^{-1}) since it represents a state of 2^{O}. However, it is also vital that this sign bit precede the number (i.e., be at the lowest order end) to facilitate the activity in the INVERTING GATES which must know the sign of a number prior to its transmission from EB to IB (ref. Sec. C-7a).

C-3i The sign then, is applied to the lowest order end of a number for the sake of the INVERTING GATES, but may, when necessary, be influenced by an end carry from the 2^{-1} position in arithmetic operations as will be seen later in Sections C-10m-o.

C-3j To obtain the COMPLEMENT of a number the procedure is: Starting with the lowest order digit and proceeding up, pass all bits unchanged up to and including the first "1" bit; thereafter invert the bits to follow.

Example:

$$\begin{array}{ll} .N & .0110100 \\ 1-.N & .1001100 \end{array}$$

An analysis of the operation is as follows:

$$\begin{array}{lr} 1-2^{-n} & .1111111 \\ \underline{-.N} & \underline{-.0110100} \\ 1-2^{-n}-.N & .1001011 \\ \underline{+2^{-n}\text{corr.}} & \underline{+.0000001} \\ 1-.N & .1001100 \end{array}$$

This may appear to be a round-about method of obtaining 1-.N, but it does not necessitate a subtractor.

C-3k The procedure of "PASS ALL BITS UNCHANGED UP TO AND INCLUDING THE FIRST "1" accomodates the subtraction of .N from all ONES and also the $+2^{-n}$ correction. "INVERSION FOLLOWING THE FIRST "1" handles the higher order portion of the subtraction from all ONES.

C-3l This scheme is know as the "2's complement system". (The "1's Complement System" does not accommodate the $+2^{-n}$ correction). When we say take the "1's complement of a number", this implies a binary point location, hence the value of the highest order numerical bit.

C-3m (The terms "'1's' complement system" and "1's complement of a number" do not necessarily mean the same thing. The G-15D obtains complements of numbers by employing the "2's complement system"; by placing the binary point to the left of the highest order bit a complement represents 1-.N hence is the "1's complement of a fractional number.") In an addition of two numbers of unlike sign, the positive number should be uncomplemented (i.e., NORMAL) and the negative number should be complemented. As a result of addition, the imaginary "1" in the 2^0 position will either remain, indicating a negative sum in COMPLEMENT form, or be changed to "0" indicating positive sum in NORMAL form. An end carry from the highest order numerical bit position (2^{-1}) will decide the fate of the imaginary "1" in 2^0 (ref. dwg. 25).

C-3n The end carry from 2^{-1} (T29 position of a word) will be added to the sign bit "2^0" (T1 position of a word) by virtue of the structure of the adders, which allow the sign position to be available for processing directly after the T29 position. Details of this will be explained under ADDERS (Sec. C-10m-o.)

C-3o Drawing 24 indicates the results of additions of two fractional numbers, a and b. Using the complementation scheme to deal with negative numbers, the chart reveals that all sums (unless they exceed a fractional value and constitute overflows) will be either in NORMAL or COMPLEMENT form and will contain the desired sign bit as CORRECTED SIGN. The results are a function of the signs and relative magnitudes of the AUGEND and ADDEND. Drawing 25 illustrates actual binary examples of the complementation and addition procedure (note that in MEMORY numbers are in terms of SIGN and ABSOLUTE VALUE: in the ADDER itself, NORMALS and COMPLEMENTS appear when appropriate).

SOURCES AND DESTINATIONS IN GENERAL

C-4a The simplest commands are those involving a transfer of information from a SOURCE (usually one memory line) to a DESTINATION (also usually one memory line). A list of SOURCES and DESTINATIONS is provided on Drawing 2. Unless DESTINATION = 31 is selected, the information content of the selected SOURCE will be copied into the DESTINATION subject to control of intervening circuits, which are under the control of the command. Except when DESTINATIONS AR+ and PN+ are selected, the original contents of the DESTINATION line are replaced by the incoming information (ref. Secs. A-7 and C-10h).

C-4b For example: if SOURCE 08, DESTINATION 16, and CHARACTERISTIC 0 are selected, this means that during the time of TRANSFER, the READING FLIP-FLOP of MEMORY LINE 08 will be coupled indirectly to the writing circuit of MEMORY LINE 16, while recirculation of MEMORY LINE 16 is blocked.

C-4c Drawing 20 illustrates the information flow from the READING FLIP-FLOP of the SOURCE line to the writing circuit of the DESTINATION line. The blocks called SOURCE SELECTOR SWITCH and DESTINATION SELECTOR SWITCH are merely portions of the individual memory lines treated collectively. As the notations suggest, activities of the blocks are a function of one or more factors contained in the command being processed.

C-4d Whenever a SOURCE is selected, from the time the STATIC PORTION of a command is read until it is replaced by that of another command, the designated SOURCE will send its information to the EB (EARLY BUS), which is nothing more than a multi-term OR gate. In case SOURCE 27, 30 or 31 is selected, more than one SOURCE line is coupled to the EB; the Boolean algebra notations on Drawing 2 indicate the logical combinations. These combinations are sometimes referred to as the LOGICAL SOURCES; commands employing them are known as EXTRACT COMMANDS. SOURCE 29 will serve to supply nothing but ZEROS to the EARLY BUS unless an extra attachment INPUT-OUTPUT REGISTER is plugged into the G-15D.

C-4e EB (EARLY BUS) information appears on the IB (INTERMEDIATE BUS) subject to modification by the INVERTING GATES. The INVERTING GATES are controlled by command information (ref. Sec. C-7b).

C-4f IB (INTERMEDIATE BUS) information can appear on the LB (LATE BUS) during the time of TRANSFER only. This time-gating by TRANSFER prevents the selected DESTINATION from being influenced at times other than the time of TRANSFER.

SPECIAL DESTINATION 31

C-4g When DESTINATION 31 is selected, the SOURCE code no longer represents a SOURCE to feed a DESTINATION (although the corresponding SOURCE line harmlessly feeds the EB). The SOURCE code designates an operation (or a choice of operations from which one may be selected by the CHARACTERISTIC code). Drawing 3 lists the operations available. (These will be described later in Sec. D.)

THE CONTROL SWITCH

C-5a When a command is read, the STATIC PORTION (T1-T13), which is concerned primarily with the routing of information, is registered in a static register consisting of 13 flip-flops. These are located in the CONTROL SWITCH (Drawing 23), and the entire contents are exhibited on the neon panel (Drawing 17).

C-5b Alteration of information in these flip-flops is possible only during the first 13 pulse periods of the READ COMMAND state defined by RC·CJ. The CX flip-flop can only be triggered when RC·CJ qualifies the AND gates controlling it; the remaining 12 flip-flops can only change state when gated CLOCK pulses are applied to them. These CLOCK pulses are the SHIFT PULSES mentioned in Section B-4. The term permitting generation of these SHIFT PULSES is actually RC·CJ. In conclusion, when a command is read these 13 STATIC FLIP-FLOPS will register T1-T13 (the STATIC PORTION) of the command and will continue to hold the information until equivalent information from the next command replaces it.

C-5c The method of feeding this information into the 13 flip-flops is shown on Drawing 22. Once any information resides in the flip-flops, an array of gates decodes the contents, delivering a variety of signals to control the activities throughout the computer. The decoding scheme is shown on Drawing 23. The output signals are stabilized after T13 of RC, and will remain stable until the end of T1 of the next RC. (During the RC·CJ period, when the flip-flops experience transition, the output signals are not interrogated.)

C-5d A combination of two output signals (Drawing 23) will precisely define a SOURCE or DESTINATION number. For instance DO·DW defines DESTINATION 02 since it is equivalent to $\overline{C6}\cdot\overline{C5}\cdot\overline{C4}\cdot C3\cdot\overline{C2}$ (i.e., 00010_2 or 02_{10} in the DESTINATION flip-flops of the CONTROL SWITCH). Such a signal combination (DO·DW) is used to allow the LATE BUS (LB) to write on LINE 02. (Note on Drawing 9 - lower left - that DO·DW·LB is the writing term of LINE 02 when selected as DESTINATION.)

C-5e Similarly, SO·SW SOURCE codes representing configuration 00010 in the SOURCE flip-flops, are suitable for controlling LINE 02 as a SOURCE. (Note on Drawing 9 that SO·SW·M2 feeds the EARLY BUS when LINE 02 is selected as SOURCE.)

C-5f By forming in this central location (CONTROL SWITCH) key signals to feed the extremities of the computer, economy in wiring and components at the extremities is achieved.

C-5g The signal DS (Drawing 23) is an example of central formation of a signal for use in many locations. DS=D7·DX·TR (i.e., a signal high during TRANSFER if DESTINATION 31 is selected). This signal "operates" the SPECIAL COMMANDS listed on Drawing 3. For instance, to perform the special command "HALT", SOURCE 16 and DESTINATION 31 must be specified (symbol: 16 ⟶ 31). When this configuration enters the 13 flip-flops and when the TRANSFER state arises the signals S4·SU·DS will be present. (S4·SU·DS = S4·SU·D7·DX·TR = CV·$\overline{\text{CU}}$·$\overline{\text{C9}}$·$\overline{\text{C8}}$·C7·C6·C5·C4·C3·C2·TR). The 3-term signal S4·SU·DS (derived from a multitude of terms) is all that is required to initiate the "HALT" command (note dwg. 30 - upper left). The same ingredients, S4, SU, & DS can be used to perform other functions. In case any one signal (such as DS) feeds many loads, it may be "boosted" by simple devices such as cathode-followers; detailed prints supply this type of information.

TRANSFER TIMING

C-6a The DYNAMIC PORTION of a command is concerned with establishing the timing control of the TRANSFER state (TR); however, how the DYNAMIC PORTION is interpreted depends to some extent upon the STATIC PORTION (which resides in the CONTROL SWITCH after T13·RC).

C-6b Drawing 19 illustrates by flow-diagram and examples the various TR timing control options. The DYNAMIC PORTION of a command holds I/D (T29) and T (T28 - T22) information. Furthermore, the factor L (WT_{RC}) has some control on timing. (The N and BP factors, also in the DYNAMIC PORTION, have no bearing on TR timing).

C-6c As the flow diagram shows, I/D and L or T control the initiation of TR regardless of the STATIC PORTION of the command. Termination of TR is a function of the configuration in the STATIC PORTION (i.e., MARK EXIT? SHIFT? MULT? DIV? NORM? S/D?). The CONTROL SWITCH supplies the necessary terms to determine mode of termination; once determined, the termination is performed by the signals occurring at the times shown on Drawing 19.

C-6d In the majority of cases, the "Imm., $\overline{\text{MARK EX.}}$, abs.", "Def., MARK EX., single", or "Def., MARK EX., double" timing will hold true. Note that in the case of SINGLE PRECISION DEFERRED commands, TR lasts one word time only (during WT = T) and in the case of DOUBLE PRECISION, DEFERRED commands, TR lasts 2 word times (T and T+1), provided T is an EVEN number. This is compatible with processing numbers of corresponding precisions.

C-6e IMMEDIATE commands can give rise to TR lasting many word times, hence are capable of processing blocks of numbers (hence, the term BLOCK TRANSFER used in programming language). It can be seen that the S/D bit has no bearing on TR timing in the IMMEDIATE cases; nevertheless, its control over the TS (TIME OF SIGN) signal remains unaffected.

C-6f The signals causing transition (at the bottom of the timing diagram) are generated in the COMMAND REGISTER and associated circuits under the control of the STATIC and DYNAMIC portions of the command. Formation of these signals will be discussed in Section C-17.

THE INVERTING GATES (EB → IB)

C-7a As was mentioned in Sec. C-4e, information may be modified in transit from the EARLY BUS (EB) to the INTERMEDIATE BUS (IB). The modification is under the control of (1) the STATIC PORTION of a command (hence the CONTROL SWITCH), and (2) the SIGNS of the NUMBERS themselves which are being transferred. Since the SIGN bit precedes the number, it is capable of controlling the circuits which process the number to follow.

C-7b The CH (CHARACTERISTIC) bits of the command, in conjunction with the S (SOURCE) and D (DESTINATION) codes control the nature of the modification in accordance with the CHARACTERISTICS TABLE on Drawing 26 upper right. By means of Boolean algebra notation, referring to CONTROL SWITCH output terms, the following transfer characteristics may be obtained:

			ABBREV.
Case I	$\overline{CX}\cdot\overline{CW}$	TRANSFER	TR
	$CX\cdot\overline{CW}\cdot\overline{S7}\cdot\overline{D7}$	TRANSFER VIA AR	TVA
Case II	$\overline{CX}\cdot CW$	ADD	AD
	$CX\cdot CW\cdot\overline{S7}\cdot\overline{D7}$	ADD VIA AR	AVA
Case III	$CX\cdot\overline{CW}\cdot(S7+D7)$	ABSOLUTE VALUE	AV
Case IV	$CX\cdot CW\cdot(S7+D7)$	SUBTRACT	SU

C-7c The CASES (I-IV) mentioned above, control the INVERTING GATES such that they transmit EB information to the IB in the manners indicated at the bottom of Drawing 26. Note that both the SIGN BIT of the number on EB and the CASE (I-IV) determine: (1) the SIGN BIT to be placed in the IB, and (2) whether or not the numerical portion of the EB number should be complemented in transit.

Drawing 26 also shows the INVERTING GATES circuit that performs these operations. CONTROL SWITCH and EB sign information controls the INVERTING GATES via gates 1-5. TS identifies the TIME of SIGN. Since the INVERTING GATES behave in accordance with TS, and TS is controlled by the S/D bit, the S/D bit has an indirect controlling effect on the INVERTING GATES.

C-7e The IS flip-flop, when set, calls for complementation of a number; when reset, the number should be transferred without being complemented (i.e., NORMAL). IS may be called the "WHETHER TO COMPLEMENT FLIP-FLOP". Note also that the RC (READ COMMAND) signal resets IS so that in the absence of any other IS control signals, no complementation will take place.

C-7f In the cases of ABSOLUTE VALUE and SUBTRACT, gate 5 prevents EB information from reaching IB at the time of the SIGN BIT. In the event that a positive number is transmitted with a SUBTRACT characteristic, gate 4 enters a "1" on the IB at sign time to provide the necessary MINUS sign. In all other cases (other than those associated with the 2-WORD LINES), the SIGN BIT on the EB will reach the IB without modification. The treatment of the sign by this circuit fulfills all of the requirements outlined on the chart at the bottom of Drawing 26.

C-7g IC is the flip-flop which scans for the first numerical "1" bit in a number to be complemented. At the beginning of a number (T2 or T2·CE), IC will be in the reset state allowing EB to reach IB without modification. If IS has been set ("WHETHER TO COMPLEMENT"), the first EB "1" bit thereafter (during TRANSFER) will set IC causing every EB bit that follows to arrive on IB inverted (i.e., after IC is set, IB = $\overline{EB}$·TS·IC). This circuit fulfills all requirements of the complementation of numbers technique described in Section C-3f-k. The sole purpose in qualifying gate 6 by TR is to prevent random setting of the OVERFLOW FLIP-FLOP (ref. Sec. C-11).

IB ⟶ LB

C-8a IB information can appear on the LATE BUS (LB) only during the time of TRANSFER (TR). If the CONTROL SWITCH yields the signal $\overline{CS}$ meaning "NOT VIA AR", this results in LB = IB·TR.

C-8b If the CONTROL SWITCH yields the signal CS this calls for a "VIA AR" characteristic and IB will be routed via the 1 word memory line AR in transit to LB.

C-8c This results in the following: (1) During the first word time of TRANSFER the original contents of AR will appear on the LB; (2) During the remaining word time(s) of TRANSFER, IB will appear on the LB delayed by one word time; and (3) During the last word time of TRANSFER, IB will be stored in AR (but will not reach the LB). This operation is a powerful programming tool.

C-8d In either case (VIA AR or NOT VIA AR) the LB will only contain information during the time of TRANSFER and consequently cannot influence DESTINATIONS at any other time.

EXECUTION OF A SAMPLE COMMAND

C-9a The performance of a sample command will now be illustrated to tie together some of the circuits and concepts discussed so far. The command is as follows:

```
I/D        T          BP        N          CH      S         D      S/D
 ↓   ┌─────^─────┐    ↓   ┌─────^─────┐   ┌^┐  ┌───^───┐ ┌───^───┐  ↓
 1   0 0·0 0 1 1 1    0   0 0 0 1 0 1 0   0 1  0 0 1 0 1 0 0 1 1 1  0
 ↑                                                                   ↑
T29                                                                  T1
```

(L = 0 0 0 0 0 0 1 = 01)

C-9b The binary configuration of this command (one of the 536,870,912 possible configurations) states the following:

(L = 01 i.e. the command is read during WT 1.)

I/D = 1 i.e. the command is DEFERRED.

T = 07 i.e. TRANSFER will be initiated at the beginning of WT 7 by T29·(T-1). For termination see S/D.

BP = 0 i.e. when this command has been performed, advance, non-stop, to the next command.

N = 10 i.e. the NEXT command is to be read during WT 10

CH = 1 i.e. ADD characteristic - complement negative numbers, pass positive numbers unchanged.

S = 05 i.e. MEMORY LINE 5 will feed the EARLY BUS

D = 07 i.e. the LATE BUS will feed the writing station of MEMORY LINE 7 during the time of TRANSFER. Also, recirculation of LINE 7 will be blocked during this time.

S/D = 0 i.e. SINGLE PRECISION: (1) TS will = T1, and (2) TRANSFER will be terminated at the end of WT 7 by T29·TR)

C-9c Result: This command, read at WT 1, causes LINE 5 to copy its contents into LINE 7 during WT 7, subject to complementation if the number in LINE 5, WT 7 is negative. The next command will be read from WT 10. The computer cycle is as follows:

RC (READ COMMAND)	WT 1	i.e. L
WTR (WAIT TO TRANSFER)	WT's 2, 3, 4, 5, 6	
TR (TRANSFER)	WT 7	i.e. T
WRC (WAIT TO READ COMMAND)	WT's 8,9	
RC (READ COMMAND - next)	WT 10	i.e. N

OPERATION ANALYSIS

C-9d (Ref. Drawing 22) During RC the command is read from a memory line previously selected and appears inverted as $\overline{MC}\cdot\overline{CG}\cdot RC$. During the first 13 pulse periods of RC, RC·CJ allows the STATIC PORTION of the command to enter the 13 STATIC FLIP-FLOPS (C1-CX). The DYNAMIC PORTION, selected by $RC\cdot\overline{CJ}$, enters the COMMAND REGISTER ADDER (discussion in Sec. C-17) to yield a trigger pulse to initiate TR at T29·(T-1).

C-9e After T13 of RC, the 13 STATIC FLIP-FLOPS in the CONTROL SWITCH stabilize yielding the following signals:

$\overline{C1}$ Means SINGLE PRECISION: (1) controls the TIMING GATES to yield TS=T1 and (2) sets up CONTROL GATES to terminate TRANSFER at T29·TR.

S1,SV SOURCE designators: qualify read-out gate of LINE 5, therefore EB = M5·S1·SV

D1,DX DESTINATION designators: qualify writing gate of LINE 7, yielding write term = LB·D1·DX. Also, during TR, the term $\overline{D1\cdot DX\cdot TR}$ is absent. $\overline{D1\cdot DX\cdot TR}$ qualifies recirculation of LINE 7. During TR recirculation is blocked since the recirculation term of LINE 7 is $M7\cdot\overline{D1}+M7\cdot\overline{DX\cdot TR}$. (DeMorgan's Law)

$\overline{CX}\cdot CW$ CHARACTERISTIC designators: These qualify gates 2 and 3 controlling the IS flip-flop in the INVERTING GATES so that TS·EB will set IS or $TS\cdot\overline{EB}$ will reset it, consequently complementing the EB number (M5·S1·SV) in transit to IB in accordance with its sign.

$\overline{CS}$ CHARACTERISTIC designator: means "NOT VIA AR". LB will = $IB\cdot\overline{CS}\cdot TR$ and will constitute the information to be written into the DESTINATION, LINE 7.

C-9f The following prints may be consulted to verify the above:

	DWG. NO'S.
"Control Switch":	23
"Timing gates" (for TS):	15
"Memory Lines":	3D296, 3D297
"Inverting Gates":	26

C-9g It may be seen that a variety of commands of this sort may be written and analyzed in a manner similar to that above.

ADDERS

C-10a The G-15D is provided with three adders (Drawing 27):
(1) AR (ACCUMULATOR REGISTER) - a single precision (1 word) adder
(2) PN (PRODUCT-NUMERATOR REGISTER) - a double precision (2 word) adder
(3) CM (COMMAND REGISTER) - a 1 word adder for special purposes. It is not available to the programmer.

C-10b The AR and PN are available for use by the programmer as general purpose adders. They are also employed to facilitate operations involved in certain special commands. (i.e. AR is used in conjunction with SHIFT and NORMALIZE, PN with MULTIPLY and DIVIDE.)

C-10c In addition to their adder capabilities, AR and PN also serve as rapid-access memory lines in that unless a NON-ZERO addend is being added to either of their contents, the contents remain unchanged (i.e., n + 0 = n) and merely recirculate in the line, available for many purposes.

C-10d Associated with each adder are three major terms: (1) the AUGEND (U), (2) the ADDEND (D), and (3) the SUM (A). In addition to these, there is the CARRY (C) term and its control system. By programming the control of U and D, the following operations are available pertaining to AR and PN:

AUGEND (U)	+	ADDEND (D)	=	SUM (A)	
U	+	(D=0)	=	U	(recirculation)
(U=0)	+	D	=	D	(read-in)
U	+	D	=	U+D	(addition)

C-10e In any case involving either U or D equal to zero, the CARRY CIRCUIT will be inactive unless triggered by some external signal.

C-10f ADDER RECIRCULATION: If programmed reference is not made to either AR or PN, whatever number happens to be in the adder (resulting from a previous operation) will recirculate as follows: A bit at the READING FLIP-FLOP will be applied to the AUGEND to which "0" will be "added" via the inactive ADDEND. This results in a SUM equal to the AUGEND. The SUM is written and will re-appear at the READING FLIP-FLOP during the next recirculation cycle.

C-10g ADDER READ-IN: When an adder is programmed as destination (but not to perform an addition), recirculation is blocked during the time of TRANSFER. That is, the information from the READING FLIP-FLOP is not allowed to reach the AUGEND input to the adder gates. Simultaneously the LB (LATE BUS) is applied to the ADDEND. This means that the writing term, A (SUM), is equal to U+D = O+LB = LB. (Hardware is illustrated on Drawing 27.)

C-10h ADDITION: An addition is based upon the assumption that the AUGEND term is recirculating in the line. If AR+ or PN+ is selected as destination, the LB will feed the ADDEND term from the appropriate SOURCE line during the time of TRANSFER. The recirculating AUGEND term will <u>not</u> be blocked. The result is that a set of gates, in conjunction with the CARRY flip-flop, will add the two terms (U and D) yielding their sum, A, which will be written on the adder's line to re-appear at the READING FLIP-FLOP during the next recirculation cycle.

C-10i SUBTRACTION: An adder may perform a subtraction by adding the negated ADDEND term. Negation of the ADDEND is performed in the INVERTING GATES which are controlled mainly by the CHARACTERISTIC (CH) bits of the command.

C-10j In loading an adder with AUGEND and ADDEND terms, the CH bits usually are selected to provide either the ADD or SUBTRACT characteristic. Below are examples, using AR, to illustrate ADDITION and SUBTRACTION programming:

<u>ADDITION:</u> $S+S' \longrightarrow A \longrightarrow$ AR write

	S	CH	D	SYMBOL
Load AUGEND	S	1	28	$S \xrightarrow{AD} AR_C$
Load ADDEND	S'	1	29	$S' \xrightarrow{AD} AR_+$

--the SUM is generated at the same time the ADDEND is entered

<u>SUBTRACTION:</u> $S-S' \longrightarrow A \longrightarrow$ AR write

	S	CH	D	SYMBOL
Load AUGEND	S	1	28	$S \xrightarrow{AD} AR_C$
Load ADDEND	S'	3	29	$S' \xrightarrow{SU} AR_+$

--the SUM (in this case DIFFERENCE) is generated at the same time the ADDEND is entered.

C-10k The ADD characteristic should be used even when loading the AUGEND. This is necessary in case the AUGEND term is a negative number. In this case the INVERTING GATES will complement the number. This is a requirement of the arithmetic.

C-10l The internal workings of the AR adder follow a conventional pattern. Addition, like transfer, is serial, bit-by-bit. Needless to say, the bits must be added starting with the lowest order bit for CARRY purposes. (This is the reason data is handled lowest order bit first and the time axis, for convenience, is usually considered going from right to left. Right-to-left time places the lowest order bit to the right. Oscilloscope horizontal-deflection plates may be reversed for this reason.)

C-10m As the reader may recall, the SIGN BIT occures at T1 of a SINGLE PRECISION number, therefore, the lowest order bit in a number is T2. An END CARRY from the T29 bit can influence the sign bit as the following paragraphs will reveal.

C-10n Prior to the addition of two numbers, the SIGN BITS (T1 of the AUGEND and ADDEND) are added, producing the UNCORRECTED SIGN bit in the SUM. No CARRY is allowed to emerge from T1 to influence the numerical portion. Furthermore, the CARRY flip-flop is unconditionally reset at the end of T1 time so it will "contain zero" at T2.

C-10o Starting with T2, the bits are added, producing the sum. Should an END CARRY emerge from T29 (highest order bit), it will be added to the UNCORRECTED SIGN during T1 of the following word time. This is because T1 of the original SUM (i.e., the UNCORRECTED SIGN) appears as the T1 AUGEND bit after the 1 word delay in the line. The final result in T1 of the SUM is the CORRECTED SIGN. The processing of the sign bit takes place twice--at the beginning of a number, then again at the end. The effect is the same as if it had all been accomplished at T1 following the highest order numerical bit (T29). An ADDITION example is provided on Drawing 29.

C-10p The principle upon which the DOUBLE PRECISION adder (PN) operates is the same as that employed by AR except that TS only occurs at T1 of EVEN word times (TE = T1·CE) and the highest order bit is T29 of the following ODD word time (T29·$\overline{CE}$). T29 and T1 at the junction of the EVEN and ODD words (T29·CE and T1·$\overline{CE}$) are numerical bits and are treated accordingly.

C-10q Drawing 27 illustrates the three adders. CM,which is not a general-purpose programmable adder, is discussed in Sec. C-17.

C-10r Note that CARRY FLIP-FLOP control differs in the AR and PN. The result is the same; only the means differ. The AR CARRY FLIP-FLOP (AC) is set or reset every single pulse period (i.e., if two or more ONES exist in U, D, and C, set AC -- otherwise reset AC). The PN (and CM) operate on the ORIGINATE-PROPAGATE-KILL basis (i.e., leave the CARRY flip-flop in its previous state unless conditions of U, D, and C call for a change.)

C-10s The latter case conserves hardware; the former case is required for the benefit of an external circuit (CONTROL GATES: in terminating TRANSFER in the case of a SHIFT command - see D-12e). Determination of sum bits is identical in AR, PN, and CM.

C-10t The CARRY set term for AR is supplemented by T29·TR·D7·$\overline{C3}$·IS·$\overline{IC}$·$\overline{AD}$, and that of PN by T29·TR·CE·DW·C6·C5·IS·$\overline{IC}$·$\overline{PD}$. These terms identify "MINUS ZERO" reaching the adder in question from the INVERTING GATES at the time of the highest order bit. By setting CARRY at this time, the CORRECTED SIGN bit will be made equal to ZERO, preventing MINUS ZERO as a number from engaging in any adder activity. Drawing 27 indicates the purpose of the various terms listed above.

(MINUS ZERO, though equal in magnitude to the normal PLUS ZERO, is incompatible with the SIGN and NUMBER scheme employed in the adders.)

C-10u To summarize ADDER activity, below are two examples showing addition of numbers of unlike sign (using 9-bit word examples for the sake of brevity):

ACCUMULATOR REGISTER (AR):

```
                        |<--- TRANSFER --->|        decimal equivalents
                        |                  |        if T29 bit represents 2^7
CARRY SET               |1   1 1     1     |                |
CARRY RESET           /1|  1     1 1   1/1 |                |
                        |                  |                v
CARRY (AC)       0 0 0/1|0 1 1 0 0 1 0 0/0 |0 0 0
AUGEND (AU)      1 0 1/1|1 0 0 1 0 0 1 0/0 |1 0 0      = +146
ADDEND (AD)      0 0 0/0|1 0 1 1 0 0 1 1/1 |0 0 0      = - 77

SUM (AA)         1 0 1/0|0 1 0 0 0 1 0 1/1 |1 0 0      = + 69
                       ^ ^                ^
  CORRECTED SIGN     T1 T29               T1   UNCORRECTED SIGN
```

PRODUCT-NUMERATOR REGISTER (PN):

```
                        |<------------ TRANSFER ------------->|
                        |                 :                   |   decimal equiv-
                        |<--- odd WT ---->:<--- even WT ----->|   alents if T29·CE
                        |                 :                   |   bit represents 2^7
CARRY SET               |1     1     1    :                   |          |
CARRY RESET           /1|  1       1      :                /1 |          |
                        |                 :                   |          v
CARRY (PC)       0 0 0/1|0 1 1 0 0 1 0 0 0:0 0 0 0 0 0 0 0/0  |0 0 0
AUGEND (PU)      0 0 0/1|1 0 0 1 0 0 1 0 0:0 0 0 0 0 0 0 0/0  |1 0 0    = +146
ADDEND (PD)      0 0 0/0|1 0 1 1 0 0 1 1 0:0 0 0 0 0 0 0 0/1  |0 0 0    = - 77
                        |                 :                   |
SUM (PA)         0 0 0/0|0 1 0 0 0 1 0 1 0:0 0 0 0 0 0 0 0/1  |1 0 0    = + 69
                       ^ ^               ^ ^               ^
  CORRECTED SIGN  T1·CE  T29·CE̅      T1·CE̅ T29·CE           T1·CE   UNCORRECTED SIGN
                  =TE                                      =TE
```

C-10v COMMAND CONTROL OF THE ADDERS: Needless to say, the AUGEND and ADDEND terms are under the control of the command which makes reference to the adder in question. The STATIC PORTION of the command, registered in the 13 STATIC FLIP-FLOPS yields the necessary output terms from the CONTROL SWITCH as follows:

AR:

(1) Block AUGEND recirculation: D7·DU·TR = DESTINATION 28 during time of TRANSFER

(2) Supply LATE BUS to ADDEND: D7·DV·LB + D7·DU·TR·LB = AD during time of TRANSFER if DESTINATION = 28 or 29. (TR in the term above is a harmless redundancy; it can be attributed to hardware conservation.)

PN:

(1) Block AUGEND recirculation: $D6 \cdot DW \cdot TR$ = DESTINATION 26 during time of TRANSFER

(2) Supply LATE BUS to ADDEND: $LB \cdot D6 \cdot DW \cdot TR \cdot \overline{CE \cdot CS} + LB \cdot D7 \cdot DW = PD$ during time of TRANSFER if DESTINATION = 26 or 30 (subject to control in case of DEST. = 26 by $\overline{CE \cdot CS}$ to be discussed in Section C-12d.

C-10w In both cases (AR and PN), other terms serve to control A and U to facilitate certain SPECIAL COMMANDS. The terms above pertain only to AR, AR+, PN and PN+ as DESTINATIONS. The READ-OUT gates follow the same pattern as general-purpose memory lines.

C-10x BLOCK ADDITIONS: It is perfectly legitimate to accumulate the sum of many numbers by allowing TRANSFER to last for several word times. In such a case, ADDENDS appear one right after the other with no "barrier" pulse periods. This means that, CORRECTED SIGN of one number will be processed in the adder at the same time as UNCORRECTED SIGN of the next number (at TS time). This, however, is permissible since the net result will be the legitimate UNCORRECTED SIGN of the second number. Drawing 29 is an example of such a case.

OVERFLOW DETECTOR (FO)

C-11a An overflow arising due to an addition may be defined as the magnitude of a sum exceeding the capacity of the adder. That is, if an adder is assumed to accommodate fractional numbers only, but as the result of adding two fractional numbers a sum is produced equal to or greater than unity, the integral portion (2^0), for which the adder has no accommodation, constitutes an overflow. Such a case can only arise from addition of numbers of like sign, which may be identified by a "0" in the UNCORRECTED SIGN bit.

C-11b An overflow may be identified by comparing certain factors including: UNCORRECTED SIGN, SIGN of ADDEND, END CARRY, etc. Drawing 27 (top) indicates the factors compared, with notes indicating the significance of the signal combinations.

C-11c The detector circuit is required to detect overflows in both the AR and PN. Furthermore, discriminating terms (from CONTROL SWITCH) are required to prevent conditions in one adder from adversely influencing the detector gates interrogating the other adder.

C-11d The term DS, which holds FE reset, prevents the OVERFLOW FLIP-FLOP (FO) from being set during a DIVIDE operation. A DIVIDE may yield legitimate overflows in PN. The DIVIDE system can set FO by "CIRCLE X" (ref. Sec. D-11ae).

C-11e The "DA-1 OVERFLOW" term can only arise if a DA-1 unit is plugged into the G-15D and is operating.

C-11f Once the OVERFLOW FLIP-FLOP (FO) is set, it will accomplish nothing unless tested by the "OVERFLOW to TEST" command (see Secs. D-14F and C-19).

This command will reset FO upon interrogation, clearing it for the next overflow. The condition of FO is exhibited on the NEON PANEL (Drawing 17)

THE TWO-WORD LINES: ID, MQ, AND PN

C-12a The two-word lines are primarily for special purposes such as MULTIPLY and DIVIDE. To facilitate these operations, certain control features are associated with information transfer to and from them. These features affect the READ-IN activity of the lines and the INVERTING GATES. Such activity is under the control of the CHARACTERISTIC bits of the associated command.

C-12b If the CH specifies anything other than TRANSFER (TR) or TRANSFER VIA AR (TVA), the two-word lines behave just as other lines do as far as READ-IN and READ-OUT are concerned. In the event that one of these lines is selected as SOURCE or DESTINATION the TR or TVA characteristic will control activity as follows:

C-12c READ-IN (Ref. Drawing 34): If ID, MQ, or PN is selected as DESTINATION with a TR or TVA characteristic, EB will be transmitted to IB; however, the SIGN BIT of each number will be blocked in transit, supplying only the absolute value to IB. IB will feed LB according to the usual rules. The SIGN BIT is fed to the IP FLIP-FLOP circuit (ref. Sec. C-13).

C-12d The LB signal will appear at the READ-IN gates of the 3 lines. CONTROL SWITCH terms will qualify the READ-IN gates and block recirculation as follows when the line in question is selected as destination:

ID: (1) TR·D6·DV blocks recirculation during time of TRANSFER

(2) TR·D6·DV·LB·$\overline{\text{CS}\cdot\text{CE}}$ writes LB in the line. The $\overline{\text{CS}\cdot\text{CE}}$ term allows LB to be written during the entire time of TRANSFER if TR, AD, AV, or SU is the characteristic. If TVA or AVA is the CHARACTERISTIC, during EVEN word times nothing is written (i.e. ZEROS); during ODD word times LB is written. This permits transfer of a SINGLE PRECISION word from an EVEN word time to the ODD (highest order) word time of a two-word line, padding the EVEN word time with ZEROS regardless of the original contents of AR. (Note that ID has only a 57 pulse delay in the line but a compensatory 1 bit delay exists in PI. PJ may be thought of as the READ-OUT FLIP-FLOP since it's output is appropriately phased.)

PN: (1) TR·D6·DW blocks recirculation of the AUGEND during time of TRANSFER.

(2) TR·D6·DW·LB·$\overline{\text{CS}\cdot\text{CE}}$ enters LB information as the ADDEND term of PN. $\overline{\text{CS}\cdot\text{CE}}$ performs the same function as mentioned in the case of ID. (If PN+ is selected as destination, $\overline{\text{CS}\cdot\text{CE}}$ does not apply; furthermore, no unusual INVERTING GATES activity will take place.)

(3) In the event that ID is selected as destination with a TR or TVA characteristic, recirculation of PN (AUGEND) will be blocked during the time of TRANSFER by TR·D6·DV·$\overline{\text{CW}}$·(CS+$\overline{\text{CX}}$).

This interaction means that loading ID with CH = TR or TVA loads ZEROS in PN. This is a programming convenience prior to a MULTIPLY command.

MQ: (1) TR·D6·DU blocks recirculation during time of TRANSFER.
(2) TR·D6·DU·LB·$\overline{CS\cdot CE}$ writes LB in the line during TRANSFER. Again, $\overline{CS\cdot CE}$ controls read-in as in the case of ID.

C-12e READ-OUT: Information will be coupled to the EARLY BUS (EB) in the same manner as that in other lines. That is, CONTROL SWITCH terms allow the READING FLIP-FLOP of the appropriate line to reach EB.

C-12f The activity of the INVERTING GATES in transmitting EB to IB will be subject to supplementary control if one of the two-word lines is selected as SOURCE and certain other conditions exist as outlined in Sec. C-13 and illustrated in Drawing 34.

C-12g Several other terms exist to control recirculation, READ-IN and READ-OUT of these lines. They are only active when certain SPECIAL COMMANDS are programmed and shall be discussed later.

THE IP FLIP-FLOP

C-13a The IP FLIP-FLOP circuit in the INVERTING GATES deals with SIGN information in cases where two-word lines are involved in READ-IN or READ-OUT and a TR or TVA characteristic is used.

C-13b In most operations related to use of the two-word lines, only absolute values of numbers are desired. SIGN information is "side-tracked" during EB to IB transmission and "stored" in the IP FLIP-FLOP when destination is ID, MQ, or PN and TR or TVA is the CHARACTERISTIC. Then, when a number in ID, MQ or PN is to be transmitted through the INVERTING GATES, the "stored" sign information is sent from IP to the IB: effectively re-assembling the SIGN and ABSOLUTE VALUE.

C-13c For example, when a MULTIPLY is to be performed, ID (MULTIPLICAND) and MQ (MULTIPLIER) are loaded with ABSOLUTE VALUES only. When the two lines were loaded, SIGN information from each number set up the IP FLIP-FLOP such that it contains the SIGN of the product to be formed by the multiplication to follow. (i.e. + x + = +, + x - = -, - x + = -, and - x - = +). When the product is formed in PN, it will be in terms of absolute value. Then when PN (PRODUCT) is transferred to memory with a TR or TVA characteristic the STORED SIGN is attached to the ABSOLUTE VALUE.

C-13d If a two-word line is the SOURCE and another two-word line is DESTINATION, the IP FLIP-FLOP will remain static. This permits data transfer among these lines to be programmed without affecting IP. Nevertheless, in such a transfer, sign information is blocked in the INVERTING GATES.

C-13e A special operation is available which appears contrary to all rules: if PN is both SOURCE and DESTINATION and TR is the characteristic, the contents of PN remain unchanged if a POSITIVE SIGN exists in IP, but will be COMPLEMENTED if a NEGATIVE SIGN is in IP. Furthermore, if a NEGATIVE SIGN is in IP, PN will acquire a "1" in its TS position. This is accomplished in the INVERTING GATES by gate 1 and gate D (dwg. 34). The operation is a convenience to the programmer, enabling him to use a PRODUCT as an AUGEND for a future ADD or SUBTRACT operation.

C-13f Drawing 34 (top) illustrates the hardware associated with the IP FLIP-FLOP and should serve to indicate the resulting activity in any particular case. The chart at the upper right illustrates how the IP FLIP-FLOP will register a PRODUCT or QUOTIENT sign according to the law of signs, provided ID is the first of the two registers to be loaded prior to a MULTIPLY or DIVIDE command. Drawing 35 shows some examples of READ-IN and READ-OUT.

THE "LOGICAL" SOURCES and SOURCE 29

C-14a The "LOGICAL" SOURCES are source numbers 27, 30, and 31. Drawing 2 defines these sources; drawing 28 shows how the appropriate terms qualify the EARLY BUS.

C-14b SOURCE 29 involves transmission of certain signals to a connector on the back of the computer into which an INPUT-OUTPUT REGISTER accessory may be plugged. These signals may be found on Drawing 23 (lower right); however, there is no need to consider them unless the accessory is used. If the accessory is not employed, the term IR will be at -20V and source 29 will constitute a convenient source of ZEROS to feed a DESTINATION when required.

THE FOUR-WORD LINES (M20 - M23)

C-15a These lines, sometimes called the RAPID-ACCESS LINES, combine the economy and versatility of drum storage with the speed of other types of storage media. Their read-in, read-out, and recirculation features resemble those of the general-purpose 108 word lines; however, LINE 23 has supplementary control features for the benefit of the INPUT/OUTPUT system.

C-15b Addressing of these lines is on the basis of "modulo 4". That is, each line has four words, numbered 0,1,2, and 3, which may be addressed during the 108 words of a drum cycle as follows:

Word 0 - word times ≅ 0 mod 4
Word 1 - word times ≅ 1 mod 4
Word 2 - word times ≅ 2 mod 4
Word 3 - word times ≅ 3 mod 4

C-15c For instance, word 1 of LINE 20 may be addressed during WT 1, 5, 9, 13, 17, etc. The word-time cycle on Drawing 6 indicates by markers the recirculation cycles of these lines. TF (i.e. T29·$\overline{\text{CE}}$·CF) is the signal indicating the end of a 4-word recirculation cycle as TO indicates the end of a 108-word recirculation cycle.

SOURCE AND DESTINATION RESTRICTIONS

C-16a LINE 0 and LINE 1: Words 107 of these two lines are reserved for the typewriter MARK and RETURN features (ref. Sec. E-9).

C-16b LINE 2 and LINE 3: Words 0-3 of these two lines are employed by the OUTPUT system. Programmed reference can be made to these words during an OUTPUT operation only under well calculated conditions.

C-16c LINE 19 and LINE 23: These lines are employed by the INPUT-OUTPUT system and should not be referred to in a program during such an operation.

C-16d LINES 14, 16, 17, 21 and 22: These lines are employed by the DA-1 (DIGITAL DIFFERENTIAL ANALYZER attachment). If such an attachment is in use and is in the "GO" state, the following terms control the lines:

LINE NO.	PERMIT RECIRC.	WRITE
14	$\overline{GO}$	M14W
16	$\overline{GO}$	M16W
17	$\overline{GO}$	M17W
21	$\overline{GO}$	ZEW
22	$\overline{GO}$	ZSW

C-16e If the DA-1 is in the "GO" state, $\overline{GO}$ is at -20V and blocks recirculation of the lines while terms arising from the logic of the DA-1 become the writing terms. If no DA-1 is attached, restrictions on use of these lines need not be considered.

THE COMMAND REGISTER (CM)

C-17a The COMMAND REGISTER is a one-word adder, the function of which is to arithmetically determine the times at which the computer should advance from state to state (i.e. RC, WTR, TR and WRC).

C-17b Unlike many computers, the G-15D does not employ "coincidence detectors" to identify word times around the drum. Instead, the COMMAND REGISTER will yield END CARRIES at times which are a function of the T and N numbers of a command. These END CARRIES will occur in accordance with the requirements of Drawing 19 (for TRANSFER TIMING).

C-17c For instance, take the example of "Def., $\overline{\text{MARK EX.}}$, single" on Drawing 19. A signal is required to initiate TRANSFER at T29·(T-1). This signal is actually T29·CC, an END CARRY out of the T28 bit of the COMMAND REGISTER, high during T29 of WT = T-1.

C-17d When the command was read during WT 3 (L = 3), the T22 - T28 bits of the command contained T = 8_{10} = 0001000_2. This information was directed (in inverted form) into the AUGEND term of the COMMAND REGISTER (i.e., CU). Simultaneously, the previously recirculating AUGEND was blocked. Refer to Drawing 22: The "block old AUGEND recirculation" term is RC·$\overline{CJ}$; the new AUGEND term is RC·$\overline{CJ}$·$\overline{MC}$·$\overline{CG}$. In the latter, $\overline{MC}$ is the INVERTED COMMAND, and $\overline{CG}$ is a control term assumed to be high unless a special command ("NEXT COMMAND FROM AR") is being obeyed (ref. sec. D-16).

C-17e Among other terms which enter CU is the T22 - T28 ("T" number) information, which is: 1110111. This is 0001000 inverted, and may be thought of as representing "-8" in the "1's COMPLEMENT SYSTEM" (inversion yields this form of complement). Assuming the lowest order bit (T22) represents 2^0 of the T number, the 1110111 configuration will represent 2^7-9, or a departure from an END CARRY (2^7 position) by a factor of 9.

If we should choose to represent the same binary configuration (1110111) in terms of the "2's COMPLEMENT SYSTEM" and "T", it is 2^7-T-1 = 128-T-1. (Ref. sec. C-3k-m)

C-17f The command was read during WT 3. At the same time that the above AUGEND (128-T-1) was being established, the NUMBER TRACK (CN) information entered the ADDEND under control of the CT flip-flop, which, in the "Def., MARK EXIT., Single" case, is high to include T3 - T28 of RC. Drawing 6 illustrates the contents of CN (the WORD-TIME reference track). From Drawing 6 it may be determined that T22 - T28 contains a factor = 'WT + 1' in all words other than WORD 107. In this case, 'WT +1' = L+1 = 3+1 = 4 = 0000100.

C-17g Drawing 22 will reveal that T21 unconditionally sets CC (CM CARRY FLIP-FLOP), rendering it high at T22 time, effectively adding a ONE to the 2^0 position of the sum. The result in the COMMAND REGISTER is as follows: CU = 128-T-1, CD = L+1, and CC = 1, yielding a sum, CA, of 128-(T-L-1), which represents a departure from 128 (END CARRY in T29) be a factor of the relative time elapsing between T29·L (i.e. T29·RC) and T29·(T-1). This relative time amounts to -(T-L-1), in this case = -(8-3-1) = -4, and will appear as the CA term in the "2's COMPLEMENT SYSTEM" form (i.e., 2^7-4 = 128-4 = 124 = 1111100_2. (The "1" in the CN 'WT+1' factor may be thought of as a $+2^{-n}$ correction factor to compensate for the negative AUGEND appearing in the "1's COMPLEMENT SYSTEM" form.)

C-17h Unconditionally, during every word time, the COMMAND REGISTER CARRY FLIP-FLOP (CC) will be high at T22 time. Also, during each word time following RC, CU = CA of the previous word time due to the one word delay in the line and the fact that the READING FLIP-FLOP supplies the AUGEND. In consequence, the SUM is increased by ONE each word time. In the example above, at T29 of the fourth word time after T29·L, the sum will represent 128-4+4 = 128-0. An END CARRY will be in CC at T29 and T22-T28 of CA will contain all ZEROS. This END CARRY, T29·CC, is the T29·(T-1) which will initiate the TRANSFER state.

C-17i This T29·CC will produce a "1" in T29·CA and will be written in the CM line. This will appear as T29·CM during the following word time and 127 word times to follow. Use of this is made in a discriminatory circuit discussed in Section C-18h.

C-17j A summary of the COMMAND REGISTER activity dealing with the "T" number using the example above is as follows:

(See next page)

```
                          T28          T22       L = 3, T = 8 (abs.)
                           |             |
                           v             v
         AUG   (CU):  X | 1 1 1 0 1 1 1 | = 128-T-1        = inv. T no. from command
WT 3     ADD   (CD):  0 | 0 0 0 0 1 0 0 | =    +L+1        = CN info. = WT+1
 (L)     CAR   (CC):  0 |             1 | =      +1        = add 1 (CC_s = T21)
         SUM   (CA):  X | 1 1 1 1 1 0 0 | = 128-(T-L-1)    = neg. rel. time remaining

         AUG          0 | 1 1 1 1 1 0 0 | = 128-4          = SUM del. one WT
WT 4     ADD            | 0 0 0 0 0 0 0 | =      0         = (no addend term)
         CAR            |             1 | =     +1         = add 1
         SUM            | 1 1 1 1 1 0 1 | = 128-3          = neg. rel. time remaining

         AUG            | 1 1 1 1 1 0 1 | = 128-3
WT 5     ADD            | 0 0 0 0 0 0 0 | =      0
         CAR            |             1 | =     +1
         SUM            | 1 1 1 1 1 1 0 | = 128-2           = neg. rel. time remaining

         AUG            | 1 1 1 1 1 1 0 | = 128-2
WT 6     ADD            | 0 0 0 0 0 0 0 | =      0
         CAR            |             1 | =     +1
         SUM            | 1 1 1 1 1 1 1 | = 128-1           = neg. rel. time remaining

         AUG            | 1 1 1 1 1 1 1 | = 128-1
WT 7     ADD            | 0 0 0 0 0 0 0 | =      0
(T-1)    CAR          1 |             1 | =     +1
         SUM          1 | 0 0 0 0 0 0 0 | = 128-0          = 0 time remaining
                     /
         [T29·CC]
```

C-17k The example above does not include the case in which the boundary between WT 107 and WT 0 is crossed in the timing cycle. In such a case, an addition of a "CORRECTIVE 20" takes place during WT 107. This CORRECTIVE 20 may be considered a factor which, when added to the 108 CARRIES added during the course of one drum cycle will yield a total of 128 or 2^7 units of addition. 128 units of addition increase the magnitude of the number in CM (T22 - T28) by a net amount of ZERO since the END CARRY effectively subtracts 128 from the 128 added. Perhaps a more convenient way of interpreting the CORRECTIVE 20 is to consider it representing 128-108 = 2^7-108 = "-108". Whenever WT 107 is passed, the next WT is 0 instead of 108. The "-108" can be thought of as "changing" WT "108" into WT 0.

C-171 The example below may serve to illustrate the function of CORRECTIVE 20:

```
                              T28         T22        L = 105, T = 2 (abs)
                               |           |
                               v           v
           AUG  (CU): X| 1 1 1 1 1 0 1 | = 128 -T-1 = 128 -2-1 = 128 -3
 WT 105    ADD  (CD): 0| 1 1 0 1 0 1 0 | =     +L+1 =   +105+1 =   +106
  (L)      CAR  (CC): 1|             1 | =-128*  +1 =-128*  +1 =   -127
           SUM  (CA): X̄| 1 1 0 1 0 0 0 | = 128 -(T-L-1)-128*    = 128-24

           AUG        0| 1 1 0 1 0 0 0 | = 128 -24          (* END CARRY subtracts
 WT 106    ADD         | 0 0 0 0 0 0 0 | =        0             128 from SUM)
           CAR         |             1 | =       +1
           SUM         | 1 1 0 1 0 0 1 | = 128 -23

           AUG         | 1 1 0 1 0 0 1 | = 128 -23
 WT 107    ADD         | 0 0 1 0 1 0 0 | =      +20 = CORRECTIVE 20
           CAR         |             1 | =       +1
           SUM         | 1 1 1 1 1 1 0 | = 128   -2 = neg. rel. time remaining
-------------------------------------------------------------------------------
           AUG         | 1 1 1 1 1 1 0 | = 128   -2
 WT 0      ADD         | 0 0 0 0 0 0 0 | =        0
           CAR         |             1 | =       +1
           SUM         | 1 1 1 1 1 1 1 | = 128   -1 = neg. rel. time remaining

           AUG         | 1 1 1 1 1 1 1 | = 128   -1
 WT 1      ADD         | 0 0 0 0 0 0 0 | =        0
 (T-1)     CAR        1|             1 | =       +1
           SUM        1| 0 0 0 0 0 0 0 | = 128   -0 = 0 time remaining
                     /
              [T29·CC]
```

C-17m Since the NUMBER TRACK contains "CORRECTIVE 20" instead of "WT+1" during WT 107, The COMMAND REGISTER will not function properly in obtaining the relative timing factor if RC takes place during WT 107. Although it is possible to calculate programming methods to compensate for this discrepancy, WT 107 is disqualified as a storage location in command lines 00 and 01 (ref. sec. C-16c). Therefore, in general, use of WT 107 as a command location is not recommended. If WT 107 is to be used as a command location the T and N numbers must be 20 higher than the values normally calculated. In cases of MULTIPLY, DIVIDE, SHIFT, and NORMALIZE, only the N number should be so treated.

C-17n During the READ COMMAND state, when the NEGATIVE RELATIVE TIMING NUMBER is initially established in the COMMAND REGISTER, a ONE may appear in the SUM term at T29 time (T29·CA). This is a function of (1) the I/D information from the command, which enters the AUGEND at T29·RC time, and (2) the possibility of an END CARRY, T29·CC, which can arise as a result of the arithmetic in this initial iteration. To assure that a ZERO will be written in the line at this time, CA is qualified by $\overline{T29}$ + CQ + CL + $\overline{CK}$, which excludes the possibility of a ONE being written at T29·RC time. In consequence, the T29·CM term cannot contain a ONE until one word time after a T29·CC·$\overline{RC}$. (Section C-18h makes reference to this particular bit.)

C-17o In the event that a command is read during one drum cycle and the T29·CC should occur during the next, the arithmetic (by virtue of the "CORRECTIVE 20") will still yield the T29·CC at WT = T-1. Also, once a command is read, the T29·CC will be generated at T29 of WT = T-1 of all drum cycles to follow until a new command replaces the old one.

RELATIVE TIMING NUMBERS

C-17p There are four SPECIAL COMMANDS (MULTIPLY, DIVIDE, SHIFT, & NORMALIZE) in which the T number is to be interpreted as a RELATIVE TIMING NUMBER rather than a reference to a particular word time. (i.e., T specifies how many word times TRANSFER should last.) These commands may be defined in terms of the following signals originating in the CONTROL SWITCH: D7·DX·S6. In these cases, when the command is read, all terms entering the COMMAND REGISTER follow the regular pattern except that the NUMBER TRACK (CN) T22-T28 contents do not reach the ADDEND term - neither during READ COMMAND nor during WT 107. The reason for this is that the T number is a relative value in the first place and it is not necessary to obtain the difference between the two absolute values of T and L to produce a relative indication of time as in the general case.

C-17q The NUMBER TRACK (CN) information is prevented from entering the ADDEND term of the CM in the RELATIVE TIMING NUMBER case by resetting the CT flip-flop (Dwg. 22) by D7·DX·S6·T21·(CL+CK) at T21 time instead of T28-- provided the computer is not in the WAIT TO READ COMMAND state. This prevents CN from becoming ADDEND to affect the T number computation which takes place during pulse periods T22 - T28 of each word time. For the benefit of the typewriter MARK and RETURN features discussed in section E-9, CL+CK (=$\overline{CL}\cdot\overline{CK}$) prevents the special CT_r during the WAIT TO READ COMMAND state. This preserves the "WORD 107" signal.

C-17r Processing of the N number in the COMMAND REGISTER is analogous to that of the T number. The N number is located in the T14 - T20 pulse periods of the word in the COMMAND REGISTER. Inversion of N to supply the AUGEND takes place by virtue of the same circuit which supplies inverted T. NUMBER TRACK information is supplied as the ADDEND during READ COMMAND and WT 107 (WT+1 and "CORRECTIVE 20") in the same manner as it is in dealing with the T number except that the NUMBER TRACK information for N takes place during pulse periods T14 - T20. A "1" is added to the lowest order place of the AUGEND by unconditionally turning the CC flip-flop on by the set term T13. An END CARRY resulting from the arithmetic, signifying the proximity of WT = N, takes place at T21 time instead of T29. The signal that is used to initiate the advancement of the computer to the READ COMMAND state is T21·CC. This will occur at T21·(N-1).

C-17s The arithmetic operations affecting the T number and N number are isolated from one another by pulse period T21. During this pulse period CC is unconditionally set by T21 (reset is prevented by $\overline{T21}$ -- among other terms), therefore, no end carry from N, or absence thereof, can have any influence upon the T number arithmetic. Drawing 33 illustrates the arithmetic in the COMMAND REGISTER that will take place in accordance with a series of three commands.

C-17t In conclusion, the COMMAND REGISTER serves the purpose of generating END CARRIES as a function of the DYNAMIC PORTION of any command, but subject to some control by the STATIC PORTION. These END CARRIES may be defined as T29·CC and T21·CC. T29·CC will occur during WT = T-1 or L+T depending upon whether the T number is interpreted as being ABSOLUTE or RELATIVE. T29·CC, in conjunction with other terms, will either initiate TRANSFER or terminate it depending on whether the command is IMMEDIATE or DEFERRED. The T21·CC will unconditionally occur during WT = N-1 and will serve to initiate the READ COMMAND state (at T29 time) providing conditions permit it. (Other aspects of the COMMAND REGISTER will be discussed in section D-6e-1.)

RC - WTR - TR - WRC CONTROL

C-18a Ref. Drawing 30: The four major states of the computer are designated by the states of three flip-flops, CL, CK, and CQ, as follows:

RC	$\overline{CL}\cdot CK\cdot\overline{CQ}$
WTR	$CL\cdot\underline{CK}$
TR	$\underline{CL}\cdot\underline{CK}$
WRC	$CL\cdot CK$

C-18b The CQ term can inhibit the RC state and in so doing creates another minor state, READ COMMAND NEXT WT. This state can only arise in the event of a TEST command (Ref. Sec. C-19). Assume CQ will always be in the reset state for the sake of the discussion to follow.

C-18c This discussion will involve starting in the READ COMMAND state and advancing through the other states. Assuming that $\overline{CL}\cdot CK\cdot\overline{CQ}$ (i.e., RC) is established by a T29 pulse, the following T29 will set CL (CL_s= RC·T29). This unconditionally terminates RC after one word time. The state to follow is a function of CK.

C-18d Had the command been an IMMEDIATE command, its T29 bit would = 0. This renders the term $\overline{CI}$ (inverted COMMAND INFORMATION) high at T29 of READ COMMAND (Dwg. 22) giving rise to $T29\cdot RC\cdot\overline{CI}$, which resets CK. Control of CK at T29 of RC is a function of the I/D bit. At this point CL and CK can assume one of the following configurations:

$CL\cdot\underline{CK}$	WAIT TO TRANSFER (if I/D bit =1)
$CL\cdot CK$	TRANSFER (if I/D =0)

C-18e In the event that TR was not initiated at T29·RC, the T29·CC END CARRY signal from the COMMAND REGISTER will initiate it (by resetting CK) at T29 of WT = T-1. Since an unwanted T29·CC could have existed at T29·RC (Ref. Sec. C-17n), a means must be provided to prevent it from prematurely initiating TR. CL·CK defines the WTR state (which does not include RC), hence T29·CC·CL·CK is a satisfactory reset term for CK in the DEFERRED command case. This accounts for the programming restriction that any DEFERRED command will involve at least one word time in WTR, hence if T = L+1, 108 word times will be spent in the WTR state.

C-18f TERMINATION OF TR: This is accomplished by resetting CL. The manner in which this is done depends upon the command--the desired results are **shown** on Drawing 19.

C-18g IMMEDIATE commands: T29·CC·TR normally terminates TR. This takes place as a result of the arithmetic in CM : T29·(T-1) for ABSOLUTE timing numbers, T29·(L+T) for RELATIVE. Other terms can also terminate TR for certain special commands:

DS·S6·SX·PM	--NORMALIZE	(Ref. Sec. D-13c)
DS·S6·SW·T29·$\overline{CE}$·AC_S	--SHIFT	(Ref. Sec. D-12e)
DS·S5·SV·T29	--MARK EXIT	(Ref. Sec. D-6f)

DEFERRED COMMANDS

C-18h SINGLE PRECISION: T29·TR·$\overline{C1}$·CM terminates TR. The T29·TR portion terminates TR after one word time, provided $\overline{C1}$ (STATIC FF: S/D bit = 0, meaning SINGLE precision) is high and CM contains a "1" at the time. CM can qualify this gate only after a T29·CC (not generated during RC) has occurred, thus preventing the gate from prematurely terminating TR in IMMEDIATE command cases. (Ref. Secs. C-17j and C-17n.)

C-18i DOUBLE PRECISION: T29·$\overline{CE}$·TR·CM terminates TR at the end of the first ODD word time after it is initiated. The purpose of the CM term is the same as in the SINGLE PRECISION case. T29·DS·S5·SV terminates TR in the MARK EXIT command case resulting in TR lasting only one word time regardless of the S/D bit (ref. Sec. D-6f).

C-18j Resetting of CL terminates TRANSFER; control of CK determines whether WRC or RC follows directly. If the TRANSFER terminating signal (TRr) occurs at T29·(N-1) time, it can, **in all but the** SHIFT and NORMALIZE cases, set CK avoiding the WRC state. Otherwise T29·(N-1) will set CK if WRC state is in effect, this state being identified by $\overline{CL\cdot CK}$.

C-18k The reader may recall that the COMMAND REGISTER "N" number END CARRY occurs at T21 time. T21·CC is not suitable for direct initiation of RC since it does not occur at T29 time; therefore, it is stored in flip-flop CJ, which qualifies gates probed either by TR_r or T29·$\overline{CK}$·$\overline{CL}$ ($\overline{CK}$·$\overline{CL}$ = WRC). CJ is normally reset by the next T13 pulse whether a T29 probing pulse sets CK or not. T21·CC will be available to control CJ on subsequent drum cycles during WT = N-1 if the first one was not used to initiate RC.

C-181 Note that T21·CC can set CJ only if the term $\overline{CH}$·CZ·$\overline{CK}$ is high. CH and CZ are the terms which allow stopping and starting of the computer (Ref. Sec. C-20) $\overline{CK}$ prevents any unwanted T21·CC generated during RC from setting CJ. Other gates capable of setting CJ will be discussed in Section D-6.

C-18m Assuming that $\overline{CQ}$ has been high during the above discussion, as soon as CK was set, the configuration CL·CK·$\overline{CQ}$ is true and the RC state exists again, starting a new cycle.

C-18n Note that when RC starts, CJ is reset by the next T13 pulse (T13·CJ·$\overline{CQ}$ + T13·CJ·TR); hence, CJ is high during the first 13 pulse periods of RC. Therefore, RC·CJ (T1 - T13) defines the STATIC portion of a command -- RC·$\overline{CJ}$ (T14 - T29) defines the DYNAMIC portion of a command.

THE "WRC NEXT WT" STATE (TEST COMMANDS) - CQ flip-flop

C-19a When a TEST COMMAND is performed, an interrogation will take place during the TRANSFER state. As a result, CQ may or may not be set in accordance with the information being interrogated. If set, the set term will occur during TR.

C-19b When T29·(N-1) yields the $\overline{CL}$·CK configuration, RC exists during WT = N, provided CQ is in the reset state; if CQ is set, $\overline{CL}$·CK·CQ exists, defining the RC NEXT WT state instead of RC. At the end of WT = N, T29·$\overline{CL}$·CK resets CQ yielding the RC state. The RC state, then, has been postponed by one word time and takes place during WT = N+1. Reset of CJ, in this case is delayed by the absence of $\overline{CQ}$ until T13 of RC to provide the desired RC·CJ and RC·$\overline{CJ}$ signals mentioned in sec. C-18n.

C-19c The CJ reset gate, qualified by T13·CJ·TR, prevents RC from taking place during WT = T+1 (instead of WT = N+1) in case certain adverse time relationships exist between the setting of CQ and the word times specified by N and T. (e.g.: An immediate NON-ZERO TEST command with L = 10, N = 20, and T = 30 finding non-zero data in word 15 would be followed by the next command read at word 31 were it not for the T13·CJ·TR gate.

C-19d The timing example of Drawing 31 should summarize state-to-state advancement activity.

C-19e The CQ flip-flop is reset by (<F>·<SA> + TAPE START)·WORD 107 + <CLEAR> to assure that the F and P keys and the turn-on cycle will cause the first command to be read at word 00 rather than word 01. (The <CLEAR> term is a redundancy since TAPE START rises during the turn-on cycle; therefore, <CLEAR> is not applied to CQ_r in some models.)

COMPUTER START-STOP

C-20a Ref. Drawing 30 (upper left): $\overline{CH}$ and CZ permit the T21·CC signal to set CJ, hence initiate the READ COMMAND state, allowing execution of the next command. Should CH be set or CZ be reset, the computer will idle in the WRC state until the $\overline{CH}$·CZ configuration is restored. Control of these flip-flops can START and STOP the computer.

C-20b MANUAL CONTROL: The right hand switch on the typewriter base is known as the COMPUTE switch. It has two compute positions--GO and BP. If it is not in either of these positions it is in the center position, $\overline{GO}$, and the computer will be stopped (in the WRC state).

C-20c With the switch on $\overline{GO}$, the $\overline{CH}$·$\overline{CZ}$ configuration will exist and RC cannot be initiated. If the switch is thrown to GO or BP, CZ will be set at T0 time and the $\overline{CH}\cdot CZ$ configuration will be established allowing RC to be initiated by T21·CC. T21·CC will occur at T21·(N-1), where N is usually a function of the last command read. (N = WT 0 can also be established by other means -Sec. E-8.)

C-20d If the switch is on GO, only a "HALT" command can stop the computer. If it is on BP, either a HALT command or a BP bit = 1 in a command can stop it. In any event, manually switching to the $\overline{GO}$ position will cause a stop.

C-20e HALT: If the HALT command is programmed, during the time of TRANSFER the following signal will be high: DS·S4·SU. This will set CH yielding CH·CZ, stopping the computer. To resume operation, the COMPUTE switch must be set to $\overline{GO}$, establishing $\overline{CH}\cdot\overline{CZ}$, then back to GO or BP, establishing $\overline{CH}\cdot CZ$.

C-20f BREAK POINT: Should the COMPUTE switch be in the BP position and the BP bit of a command =1, T21·CI·RC·<BP> will become high, setting CH. (CI is command information; ref. dwg. 22.) Note that this term can only become high if <BP> is present; BREAK POINTS are ignored if the COMPUTE switch is on GO. <BP> must qualify the CH_s gate prior to starting the computer so the first BP's in a program cannot be bypassed; for this reason, the <BP> contact of the COMPUTE switch must be grounded prior to the [<GO> + <BP>] contact.

C-20g "I" key operation: This provides a means of performing one command then stopping and is sometimes called the SINGLE CYCLE operation. See Section E-5.

C-20h (<F>·<SA> + TAPE START)·WORD 107 is applied to CH_s to assure that the computation cycle is stopped in case the COMPUTE SWITCH is accidentally left on GO or BP during the turn-on cycle or when the F or P key is being used.

C-20i It is of historical interest to note that in earlier documents the "true" side of the CH flip-flop used to be opposite to its present location. This meant that the computer was stopped by resetting CH rather than setting it and the HALT neon lit when CH was reset. There is no change in theory of operation; only the nomenclature was altered.

DESTINATION 27 - TEST (NON-ZERO)

C-21 Any SOURCE may feed DESTINATION 27. If during the time of TRANSFER one or more ONES appear on the LATE BUS, the test will be met, CQ will be set, and the next command will be read at WT = N+1 (instead of N). The term LB·D6·DX in this case will set CQ (ref. dwg. 30) causing a one word delay in the RC state.

SECTION D

THE SPECIAL COMMANDS

This section requires a reasonable understanding of previous sections, although back references are given to assist the reader's memory. Little description is devoted to optional auxiliary equipment which is associated with certain commands, since such equipment is a study in itself and is documented in other literature.

THE SPECIAL COMMANDS-GENERAL

D-1a The SPECIAL COMMANDS are those in which DESTINATION 31 is selected and the SOURCE number, and sometimes CHARACTERISTIC, define the operation to take place during the time of TRANSFER. Drawing 3 lists the operations.

D-1b SPECIAL COMMANDS with a SOURCE number less than 16 all initiate INPUT-OUTPUT operations and will be discussed in sec. F. In this section SPECIAL COMMANDS with SOURCE numbers equal to or greater than 16 will be discussed.

D-1c "DS": A CONTROL SWITCH (dwg. 23) signal common to all SPECIAL COMMANDS is DS ("DESTINATION SPECIAL"). DS = TR·D7·DX, hence is a special case of TRANSFER. This signal ordinarily activates all SPECIAL COMMANDS by qualifying the appropriate gates. This will become apparent as SPECIAL COMMANDS are analyzed.

SOURCE 16: HALT

D-2 DS·S4·SU sets CH, causing the computer to idle indefinitely in the WRC state after TR is terminated. (ref. Dwg. 30-upper left and sec. C-20e).

SOURCE 17: RING BELL, ETC.

D-3a SOURCE 17, RING BELL: DS·S4·SV (see dwg. 23 right) is applied to the grid of a relay puller tube (V12 in the LOGIC PANEL - RELAY SECTION - dwg. 64) during the time of TRANSFER. If TRANSFER lasts 108 word times (approximately 30 milliseconds), RY1 will pull in and its points will complete the circuit to energize the bell solenoid. The command should be immediate with T = L+1. Considering the electro - mechanical requirements of the hardware associated with the bell, three drum cycles should elapse between repeat executions of this command.

D-3b SOURCE 17·CH=1, RING BELL and ⟨MAN PUNCH⟩ TO TEST: (dwg. 30-left) DS·S4·SV·① arises during TR, interrogating the ⟨MAN PUNCH⟩ signal. ① = $\overline{CX}$·CW (char. =1). ⟨MAN PUNCH⟩ is a signal which is high if the PUNCH SWITCH on the typewriter is in the ON position. If the punch switch is ON, DS·S4·SV·①· ⟨MAN PUNCH⟩ will set CQ and the next command will be read from WT =N+1. The bell will ring if TR lasts 108 word times; if TR is programmed to last only one word time the bell will not ring but the test will take place. (1 word time = approximately 270 microseconds, which is insufficient to activate the bell circuit relay or the bell).

D-3c SOURCE 17·CH=2, RING BELL AND START INPUT REGISTER: DS·S4·SV·② is coupled to a connector into which an INPUT-OUTPUT REGISTER accessory may be plugged (see dwg. 23-right). This signal, high during TR, is for the purpose of controlling the above accessory, but can be used to control other types of external accessories.

D-3d SOURCE 17·CH=3, RING BELL AND STOP INPUT REGISTER: DS·S4·SV·③ is coupled to a connector, etc. (see D-3c).

SOURCE 18: TRANSFER M20·ID to OUTPUT REGISTER

D-4 This operation supplies M20·ID to an output connector during the time of TRANSFER, providing DS·S4·SW is present. The signal to output is DS·S4·SW·M20·PJ, where PJ is ID's output flip-flop. In addition to this, WRITE PULSE is supplied to the connector, by itself and also qualified by DS·S4·SW·PJ. If the INPUT-OUTPUT REGISTER accessory is connected, the above hardware (dwg. 23-right) supplies it with LINE 20 data and WRITE PULSES under control of ONES in the ID REGISTER.

SOURCE 19: DA-1 CONTROL

D-5 Ref. Dwg. 23 - right: DS·S4·SX·④ and DS·S4·SX·① are sent to a connector into which a DA-1 (DIGITAL DIFFERENTIAL ANALYZER) may be plugged. SOURCE 19·CH=0 STARTS the DA-1; SOURCE 19·CH=1 STOPS the DA-1.

SOURCES 20 and 21: SELECT COMMAND LINE, MARK EXIT, AND RETURN EXIT

D-6a These commands frequently are programmed as a pair and serve to set up new command lines in addition to other operations.

D-6b Commands are read during the RC state from one of eight possible COMMAND LINES (00, 01, 02, 03, 04, 05, 19, and 23). The particular line which supplies the commands is a function of three flip-flops: CD1, CD2, and CD3. The three flip-flops, once set up in a particular configuration, will remain in that configuration until changed by programming or by manual control. Therefore, commands will emerge from the same line until another line is established by changing the configuration of the CD flip-flops.

D-6c The CD flip-flops are shown on Drawing 22-bottom. Above them are eight gates supplying $\overline{MC}$, the inverted command information from the false side of the selected command line. Which gate is qualified, hence which line supplies commands, is a function of the CD's.

D-6d When either SOURCE 20 (RETURN EXIT) or SOURCE 21 (MARK EXIT) is programmed, DS·S5·$\overline{C}$8 appears during TRANSFER, setting the CD's in accordance with the contents of CX, CW, and C1, which are the CHARACTERISTIC and S/D bits in the 13 STATIC FLIP-FLOPS. Hence a new command line may be established by the CH and S/D bits as shown on Drawing 32 (lower left). Other signals originating in switch contacts can control these flip-flops (see secs. E-6, E-7, E-11d and G-2m). In addition to the command line selection function, these two commands perform other functions as follows:

D-6e SOURCE 21, MARK EXIT: (Ref. dwg. 32). This command, during the time of TRANSFER, blocks recirculation of the COMMAND REGISTER AUGEND term during pulse periods T2-T13, effectively loading this portion of CM with ZEROS. T1·DS·S5·SV sets CJ; T13·CJ·(TR+$\overline{CQ}$) resets it. CJ·DS·S5·SV blocks CU. The reader may recall that CJ has other functions (sec. C-18k-n); this does not prohibit its use in this capacity since the different functions are isolated by a barrier of time.

D-6f If the MARK EXIT command is written as deferred, TRANSFER takes place during WT=T only, regardless of the S/D bit of the command.

(TR_r = DS·S5·SV·T29-ref. sec. C-18g and dwg. 30.) This allows the S/D bit to equal 1 or 0 (in accordance with the desired number of the new command line to be selected) without influencing the TRANSFER state.

D-6g Since CU was rendered equal to zero during TR, and TR occurred during WT = T, then CU = 0 at WT = T. T1 unconditionally sets the COMMAND REGISTER CARRY FLIP-FLOP (CC) every word time adding a ONE to the lowest order bit (T2) of this newly established AUGEND. Hence, during WT = T, the sum (CA) equals 1. During the next 107 word times 107 carries will be added to this number (CC_s=T1) and at WT=107 a CORRECTIVE 1940 will enter the ADDEND (CD) from the NUMBER TRACK (CN) yielding a total of 2048. This sum equals 2^{11} and will appear as (1) all ZEROS in the T2-T12 positions of the COMMAND REGISTER line and (2) an END CARRY at T13 time of WT = T-1 of the drum cycle following the MARK EXIT command. Furthermore, this T13·CC will exist at WT = T-1 during every drum cycle thereafter until another MARK EXIT command establishes a new "MARK".

D-6h SOURCE 20, RETURN EXIT: (ref. dwg. 30-top) When this command is read, during the time of TRANSFER it will perform its command line switching function. However, when it comes to establishing the next READ COMMAND state, an additional gate can set CJ to permit the next T29 pulse (TR_r+T29·CL·CK) to initiate RC. This new gate is qualified by T13·CC·CJ·D7·DX·S5·SU and allows the T13·CC, established by the last MARK EXIT command, to initiate RC instead of the usual T21·CC. That is, the next command will be read during WT = T of the last MARK EXIT command instead of WT = N of the RETURN EXIT command.

D-6i The above will take place provided the MARK EXIT and RETURN EXIT commands are constructed as illustrated on Drawing 32.

D-6j If the previous MARK EXIT command had been written as IMMEDIATE, the T13·CC would have occurred during WT = L of the MARK EXIT command (as a function of TR), hence the command following the RETURN EXIT would be read during WT = L+1 of the MARK EXIT command. (The T number of an IMMEDIATE MARK EXIT command can be anything; however TR will last only one word time since TR_r=T29·DS·S5·SV.)

D-6k The T13·CC will initiate RC following a RETURN EXIT only if the T21·CC does not initiate it first. Relative timing of these two CARRIES will determine the location of the next command. When N of the RETURN EXIT command = L+1, T13·CC initiating RC prior to T21·CC is inevitable.

D-6l A RETURN EXIT command should not have its BP bit =1. This will serve to prevent T21·CC from initiating the next RC state (see sec. C-20); however, it will not prevent T13·CC from doing so. If T13·CC initiates RC, the state of CZ and CH will prevent T21·CC established by the new command from initiating the RC state to follow, effectively postponing the stop by one program step. Also, if relative timing of T13·CC and T21·CC is such that T21·CC is intended to precede T13·CC and initiate RC during WT = N of the RETURN EXIT command, the state of CZ and CH will block T21·CC to initiate RC regardless of sequence. This would mean that a program could proceed in one way with the COMPUTE SWITCH on "GO" and another with the COMPUTE SWITCH on "BP". SINGLE CYCLING (sec. E-5) will also give rise to this problem for the same reason. The solution is: 1) do not place a BP in a RETURN EXIT command and 2) do not SINGLE CYCLE a RETURN EXIT command unless fully aware of all possible results.

SOURCE 22: SIGN OF AR TO TEST

D-7 If this command is programmed, it will cause the next command to be read from WT = N+1 if a negative number exists in AR. This is accomplished by DS·S5·SW·T1·AR setting CQ (ref. dwg. 30-left and sec. C-19).

SOURCE·23 CH=0: CLEAR MQ, ID, PN and IP FLIP-FLOP

D-8 DS·S5·SX·④ blocks recirculation of the three two-word lines and resets the IP flip-flop (ref. dwg. 36). TRANSFER must last for two word times to block recirculation of the lines for a complete recirculation cycle. Exactly when these two word times take place makes no difference.

SOURCE 23·CH=3: PN·M2 to ID and PN·$\overline{M2}$ to PN

D-9a (Ref. dwg. 36) This command causes two separate concurrent "extract" operations:

1) PN·M2 to ID: This requires blocking ID recirculation and application of PN·M2 to the ID writing circuit. DS·S5·SX·③ accomplishes the former, DS·S5·SX·③·M2·PP accomplishes the latter. (PP is the READ FLIP-FLOP of the PN register).
2) PN·$\overline{M2}$ to PN: This operation can be described as blocking recirculation of PN if M2 is high. DS·S5·SX·③·M2·PP accomplishes this.

D-9b The operations performed by the command facilitate efficient DECIMAL--BINARY conversion routines, but may be used to advantage for sorting purposes in any program.

SOURCE 24 : MULTIPLY

D-10a MULTIPLY may be considered an automatic sub-routine built into the computer hardware. The purpose of the sub-routine is to form a fractional product (in PN) from a fractional multiplicand (in ID) and a fractional multiplier (in MQ). The binary point, for this purpose, must be considered to the left of the highest order bit (T29·$\overline{CE}$) in these registers.

D-10b The sub-routine involves the addition (accumulation) of: $(ID_O \times 2^{-1}) \times 2^{-1}$ of MQ_O + $(ID_O \times 2^{-2}) \times 2^{-2}$ of MQ_O ··+ $(ID_O \times 2^{-n}) \times 2^{-n}$ of MQ_O, these values being accumulated in the PN adder to form a product. The hardware is shown on Drawing 36 and an example on Drawing 37.

D-10c In order to obtain $(ID_O \times 2^{-n})$, a means of shifting the contents of ID to the right one bit each recirculation time is provided (i.e. ID shifted right one bit = ID x 2^{-1}, two bits = ID x 2^{-2}, three bits = ID x 2^{-3}, etc.).

D-10d The means of shifting ID is shown on Drawing 36 (left). The PJ flip-flop reads the contents of ID one recirculation time after it is written

(57 +1 = 58 pulse delay) and is considered to be the "OUTPUT FF" of the line. If PJ is the writing term of ID (normal recirculation), information in the line is synchronized such that every recirculation time it is written in the same place relative to the time reference. If, however, PI is coupled as the writing term instead of PJ, information will be written one pulse-period in advance, meaning that bits will be recirculated out of phase with the time reference resulting in the information being advanced one bit each recirculation cycle.

D-10e This principle, known as PRECESSION, effectively shifts the number in ID to the right (relative to this right-to-left time axis) one bit every two word times. This particular case may be referred to as a "ONE BIT PRECESSION TO THE RIGHT".

D-10f Note that $\overline{TE}$ qualifies PI when it is the writing term. This means that during TE (sign position of double precision words), nothing but a ZERO can be "written". 57 pulse periods later (T29·$\overline{CE}$), PI will read this ZERO. In effect, this $\overline{TE}$ qualifying term inserts ZEROS into the highest order end of the word as it is shifted to right; bits at the lowest order end of the numerical portion are lost as the number shifts right. Drawing 37 illustrates this in terms of a ten-bit example.

D-10g Note in the example on Drawing 37 that even though ID_o and MQ_o contain single precision numbers, the PRODUCT acquires numerical bits below the lowest order numerical bit position of PN_{odd} (i.e. T2·$\overline{CE}$). If a subsequent single precision command transfers only PN_{odd} to memory, the PRODUCT sign bit stored in IP will be substituted for the numerical bit in T1·$\overline{CE}$ of PN in the INVERTING GATES during TRANSFER (ref. D-10p and Drawing 34).

D-10h As a result of the PRECESSION circuit mentioned above, ID times 2 to reducing powers is available to be accumulated in the PRODUCT REGISTER (PN) in accordance with the MULTIPLIER (MQ) data. A means must be provided to interrogate the bits in MQ corresponding to these powers of 2 for the purpose of controlling the additions of ID x 2^{-n} in the PN register.

D-10i The means of interrogating MQ bits, starting with the highest order bit (2^{-1}), also involves precession. Ref. Drawing 36: The contents of MQ can shift to the left by virtue of delay flip-flop PQ. During a multiplication, normal recirculation of MQ is blocked and another recirculation path including a one bit delay (provided by PQ) is substituted. The result of this is that information from the reading station is re-written after a 59 pulse delay, effectively shifting it one bit position to the left relative to the right-to-left time axis each recirculation cycle. This may be called a "ONE BIT PRECESSION TO THE LEFT".

D-10j $\overline{TE}$ qualifies the writing term when this precession path is connected. This means that at TE time a ZERO is written each recirculation cycle, effectively inserting zeros in the lowest order end of the number each time it shifts to the left one bit. Furthermore, this means that as the number shifts to the left, the highest order bit is shifted "out of the end" of the line and lost.

D-10k The PM FLIP-FLOP monitors the highest order bit of MQ (at T29·$\overline{CE}$ time) at the end of each recirculation cycle. Timewise, PM will register, for a 2 word time duration, each MQ (multiplier) digit starting with 2^{-1}.

D-10l Whenever the highest order MQ bit is a ONE after a shift, ID x 2^{-n} will be applied to the ADDEND of PN, hence accumulated in the product (PD= PM·PI·DS·S6·$\overline{\text{C8}}$·$\overline{\text{C7}}$·$\overline{\text{TE}}$). In the event of a multiplier bit equal to ZERO, no addition takes place. After the last (lowest order) MQ ONE bit has performed its ID x 2^{-n} addition function, the product is complete. If a non-ZERO number existed in PN prior to the multiplication, it would be included in the product. Sometimes this is convenient; at any rate, programming will determine the initial contents of PN.

D-10m The "T" number in a multiplication command is interpreted as being RELATIVE (sec. C-17p). If T = 114, this allows 57 MQ bits to be interrogated (114 WT's = 57 two-word recirculation cycles) hence will form a complete product in accordance with a double precision multiplier up to and including the lowest order MQ numerical bit (T2.CE). If the multiplier is a single precision number, only 28 numerical MQ bits need be interrogated and T may be equal to 56 with no loss in product precision. Furthermore if the multiplier is a constant, interrogation of MQ bits need only proceed to include the lowest order ONE bit. (e.g., if the lowest order one bit is in the 2^{-5} location, only 5 iterations are necessary and T may equal 10 with no loss in product precision - each iteration requires 2 word times, the recirculation time of a two-word line.)

D-10n The MULTIPLY command must be located in an <u>odd word time</u> so that the first word time of TRANSFER takes place at an even word time, the beginning of a recirculation time of a two-word line. The command must be IMMEDIATE and T must be an even number to assure termination of TRANSFER at the end of a two-word line recirculation cycle (odd word time). The CH and S/D precision bits can contain anything since they do not control anything associated with multiply; normally these bits are ZEROS. (SOURCE and DESTINATION contents obviously must be 24 and 31 respectively.)

D-10o Round-off: Should the total number of significant MQ and ID bits exceed 57, the lowest order end of PN will be inaccurate in that bits from ID x 2^{-n} will be lost from the lowest order end of ID. This form of round-off is not conventional and should be recognized by the programmer.

D-10p The sign of the product is not included in PN but is held in the IP FLIP-FLOP. When the product is transferred from PN to memory with a TR or TVA characteristic, the sign in IP will be attached to the absolute value of the product (sec. C-13). A multiplication overflow is impossible since the product of two fractional numbers will always be fractional.

SOURCE 25 : DIVIDE

D-11a DIVIDE, like MULTIPLY, is an automatic sub-routine built into the computer hardware. The sub-routine obtains the ratio of two numbers and inserts it, bit-by-bit, into the MQ. The operation is: NUMERATOR (PN) ÷ DENOMINATOR (ID) → QUOTIENT (MQ). The quotient may only approach 2 in magnitude (i.e., 1.11111---1). For this reason the ratio of PN to ID must be less than 2 (i.e. PN/ID<2 or PN<2ID is a requirement of the input data).

D-11b A DIVIDE command may be written in several ways as far as TRANSFER timing is concerned depending upon the form in which the quotient is desired. These are outlined on Drawing 40 - lower right.

D-11c Cases involving T = 57 or 116 are most frequently used since the binary point exists in a conventional location. PN must be $<$ ID to assure that the quotient (MQ) will be fractional; an integral bit (2^0) constitutes an overflow in these cases and will be detected (sec. D-11ae).

D-11d Cases involving T = 55 or 114 have their programming virtues. In these cases, if PN/ID$\geq$2, a useless quotient will result; furthermore, no overflow indication will exist (i.e., FO FLIP-FLOP). The virtue of these cases exists in the accommodation of one integral bit (2^0).

D-11e In all cases the quotient information replaces the original contents of MQ as indicated on Drawing 40. The final sign bit (T1·CE) will = 0 and the quotient will be in terms of absolute value.

D-11f In all cases command information must include I/D = 0, CH = 1, S = 25, and D = 31 to define the command. BP and N will be programmed as in any other command. LOCATION of the command must be an ODD word time.

D-11g The S/D bit can contain anything since other terms in the STATIC PORTION of the command assure that TS = TE regardless of the S/D bit (see dwg. 15: $\overline{S6 \cdot SV \cdot DS}$, low during DIVIDE, forbids generation of TS at T1·CE time even though $\overline{C1}$ calls for it. $\overline{C1}$ means S/D bit = 0 = single precision.) If the DIVIDE command calls for only a single precision quotient, the iterations involved in its formation are nevertheless dealing with double precision registers and the timing must be controlled accordingly.

D-11h The division algorithm employed by the G-15 is essentially a standard one with certain deviations to minimize the electronic control circuitry. Drawings 38 and 39 illustrate the evolution of the system employed. Drawing 38 figure (A) illustrates "long division" in binary, using the method analagous to the decimal system taught in grade school. At the extreme left of figure (A) is the abbreviated version of the division; to the right is the same example, iteration-by-iteration. Insignificant ZEROS are shown as solid lower-case ZEROS and account for 12 binary places. Associated with this example are algebraic notations and orders of magnitude as an analysis of the proceedings.

D-11i As the notations suggest, the denominator (ID) times 2 to decreasing powers is subtracted from the numerator (PN) to achieve minimum positive remainders. Whenever ID x 2^{-n} can be subtracted from the previous remainder rendering a positive remainder, a ONE representing 2^{-n} is put in the quotient. Whenever ID x 2^{-n} cannot be subtracted from the previous remainder without yielding a negative remainder, the subtraction is by-passed (shown as subtraction of ZEROS), and a ZERO appears in the quotient corresponding to 2^{-n}.

D-11j This system depends upon an inspection to determine the relative magnitudes of ID x 2^{-n} and the remainder from which it will be subtracted in order to determine the feasibility of the subtraction to follow. This cannot be done by the computer without actually performing a subtraction to determine whether the remainder will be positive or negative.

D-11k Therefore, the algorithm in figure (B) is considered. This does not involve inspection of magnitudes followed by subtraction or no subtraction, but instead involves both additions and subtractions of ID x 2^{-n} from PN. The remainders are both positive and negative. The sign of each remainder determines the following:

+ remainder : 2^{-n} quotient bit = 1
next operation will SUBTRACT ID x 2^{-n-1}

- remainder : 2^{-n} quotient bit = 0
next operation will ADD ID x 2^{-n-1}

D-11l In this case, the same quotient is developed as in the first case (fig. A) and the positive remainders are the same. This may be confirmed by inspection of the algebraic notations.

D-11m Note that once an all ZERO remainder is established, all remainders to follow must be negative since $[- \text{ID} \times 2^{-n} + \text{ID} \times 2^{-n-1} + \text{ID} \times 2^{-n-2} - - - + \text{ID} \times 2^{-n-m}]$ can never be positive; hence all quotient bits developed following a ZERO remainder will be ZEROS.

D-11n The reader will note that in both figures (A) and (B) insignificant ZEROS account for a great many of the bit locations. Therefore a system illustrated in figure (C) will be considered. This system involves multiplying each remainder by 2 and not multiplying ID by 2 to reduced powers. Hence the denominator, unchanged, is either added to or subtracted from 2 times the preceding remainder. This keeps orders of magnitude of the factors approximately in the same range without altering the signs of the remainders. The only difference between remainders in figures (B) and (C) is a scaling factor which is compensated for by adjustment of the ID scaling factor. In this system, bit locations need not be allotted in wholesale quantities to insignificant ZEROS. This implies more efficient use of registers hence greater precision capabilities of registers of fixed length.

D-11o This last algorithm is the basis of the G-15D division system. However, the G-15D does not have a subtractor. Furthermore, a sign bit must be considered. Therefore the algorithm must be adapted to the arithmetic system in use and the circuits available.

D-11p Drawing 39, fig. (D) is a proposition for performing the algorithm illustrated on Drawing 38, fig. (C), but performing subtractions by adding COMPLEMENTS of ID. A sign bit is located to the left of the highest order (2^0) bit position. (The sign bit of the two-word registers occurs at T1·CE which follows directly the highest order numerical bit, T29·$\overline{\text{CE}}$).

D-11q An integral (2^0) position is allowed in the example on Drawing 39, figure (D) to accommodate shifted negative remainders (i.e., r x 2) which can approach 2 in absolute value (see dwg. 38, fig. (C) - orders of magnitude). Note that in the proposition on Drawing 39, fig. (D), the 2^0 bit and sign bit of each unshifted remainder are always alike. It can be proved that this will always be the case (considering the fact that signs of augend and addend are opposite and the absolute value of any augend cannot exceed twice the absolute value of its associated addend).

D-11r Since the integral and sign bits in the unshifted remainders will be alike, they can occupy the same bit position as shown in Drawing 39, figure (E). In terms of the PN, where this arithmetic takes place, T29·CE is the location of the highest order fractional bit (2^{-1}) and T1·CE (= TE = TS) is the location of the "combination" sign and integral bit.

D-11s For the sake of division analysis, consider the sign bit of a number as following the number rather than preceeding it. There is no sign position at the beginning of the number associated with that number. Consider the iteration cycle displaced from a standard two-word recirculation cycle by a factor of one bit period (i.e., the iteration cycle starts at T2·CE, not T1·CE). Drawing 39, figure (G) illustrates this concept.

D-11t Note that in Drawing 39, figure (E), when the remainders are shifted left one bit (multiplication by 2), they still appear in the same form as they did prior to shifting (i.e. NORMAL or COMPLEMENT) except in the cases noted "N". These cases result when a negative remainder in complement form exceeds a fractional magnitude when shifted and an overflow exists. The shifted remainders in these cases resemble numbers in NORMAL form since their sign bits = 0; they are nevertheless COMPLEMENTS. When the addend is applied to such a number, END CARRY behavior compensates for the overflow and yields the next remainder in correct form.

D-11u This system (fig.E) would be that used in the computer except that the sign bit associated with each unshifted remainder must be reversed. This SIGN BIT decides whether an addition or subtraction will take place in the next iteration (sec. D-11k) by controlling the IS FLIP-FLOP in the INVERTING GATES. The contents of ID, being applied to the EARLY BUS, will arrive on the LATE BUS (hence the PN addend) either in COMPLEMENT or NORMAL form in accordance with the orientation of IS.

D-11v At TE (sign) time, the remainder sign (TE·PA) is applied to the EARLY BUS and consequently controls IS in accordance with the rules of the ADD characteristic (programmed in the command). This means a "1" in the sign bit of the sum (PA) will set IS, complementing ID for use as the next addend, hence performing a subtraction. This is the exact opposite of the desired results. Therefore, the sign bit of every remainder is purposely reversed. This is accomplished by applying the wrong sign bit to each addend (i.e. 1 for +, 0 for -). Drawing 39, fig. (F) illustrates this arrangement.

D-11w Fig. (G) illustrates fig. (F) in terms of the time axis. This six-bit example illustrates the activity in the 58 bit register actually used. Note that the first iteration unconditionally involves a subtraction since the EB bit at the first pulse period during TRANSFER is a ONE as a function of the starting data.

D-11x During every iteration the IS flip-flop in the INVERTING GATES will remain set or reset as a result of the sign of the previous remainder. The setting of IS controls the following items:

Positive remainder, IS is high.

1) Supply the numerical portion of ID, COMPLEMENTED, to PN as ADDEND (for subtraction)
2) Supply a "0" as sign bit to the PN ADDEND at TE time.

3) Allow a "1" to be retained by MQ as quotient bit.

Negative remainder, $\overline{IS}$ is high

1) Supply the numerical portion of ID, UNCOMPLEMENTED, to PN as ADDEND (for addition)
2) Supply a "1" as sign bit to the PN ADDEND at TE time.
3) Do not allow a "1" to be retained by MQ as quotient bit (i.e. quotient bit = 0)

D-11y Regardless of the remainder of the final iteration prior to termination of TRANSFER, a "1" is entered as the lowest order quotient bit. This is known as the PRINCETON ROUND-OFF and has its mathematical virtues.

D-11z Drawing 41 illustrates the 6-bit example used during this evolution in terms of actual signals associated with the circuits involved in a division. A block diagram of the DIVIDE circuits is shown on Drawing 40.

D-11aa ID supplies the EARLY BUS by virtue of S6·SV which is the same term that allows ID to feed EB when ID is programmed as SOURCE, and DESTINATION is not 31.

D-11ab QUOTIENT: The MQ precesses to the left during a divide operation. During every cycle a "1" is inserted in the line at T2·CE time (lowest order bit). 2 word times later at T2·CE time, the state of IS (a function of the previous remainder's sign) determines whether or not this "1" can reach the writing station at T3·CE time and be retained as a quotient bit = 1.

D-11ac Since division is terminated by the TRANSFER terminating pulse, which occurs at T29, the last "1" written at T2·CE time will not be blocked; this provides the PRINCETON ROUND-OFF.

D-11ad At the conclusion of a division the MQ will contain the absolute value of the quotient with a precision which is a function of the number of iterations, hence the T number. The sign bit, now again considered at the low end of the number will contain ZERO. The actual quotient sign is held in IP and will be applied to the number when it is transferred to memory (sec. C-13).

D-11ae OVERFLOW: If at the time of TR_r, (TR reset pulse) a "1" exists in the reading flip-flop of MQ, it represents a 2^0 quotient bit = 1 and constitutes an overflow in the cases when a fractional quotient is called for by programming T = 116 or 57. The reason that a single precision division is programmed to last 57 word times, rather than an even number, is so that the TR_r pulse will occur at T29·CE time (rather than T29·$\overline{CE}$)and will interrogate the state of the appropriate bit. In the event of an overflow, ⓧ (=DIVIDE·PR·TR_r) will set the OVERFLOW FLIP-FLOP (FO).

SOURCE 26 : SHIFT MQ LEFT AND ID RIGHT

D-12a Ref. Drawing 36: This command performs the SHIFTING operations by virtue of the same circuits employed by MULTIPLY (which also shifts MQ left and ID right). CONTROL SWITCH terms control precession of ID and MQ during the time of TRANSFER.

D-12b Also high during TRANSFER is the term $T1 \cdot \overline{CE} \cdot DS \cdot S6 \cdot C8 \cdot$ ④, which turns on the CARRY FLIP-FLOP (AC) of the ACCUMULATOR REGISTER (AR). This results in adding a ONE to the lowest order numerical bit position (T2) of AR during every ODD word time of the time of TRANSFER if characteristic is 0. The number in AR will be "incremented" once every two word times and will count the number of bit positions the contents of ID and MQ are shifted.

D-12c Assume that n is the desired number of bit positions of shift to be imposed upon ID and MQ. If this number is established in AR (prior to the shift command) as a negative number (in complement form) it will appear as $1 - n \times 2^{-28}$.

D-12d Once the TRANSFER state of the SHIFT command is initiated, at the beginning of an EVEN word time, precession will take place in ID and MQ. At the end of each recirculation cycle of the two-word lines ($T29 \cdot \overline{CE}$) the sum in the AR will have been increased by 2^{-28}. That is, after the contents of ID and MQ have shifted one bit, AR will contain $1 - n \times 2^{-28} + 2^{-28} = 1 - (n-1) \times 2^{-28}$. After 2n word times of TRANSFER, the contents of ID and MQ will have been shifted n bit positions and AR will contain $1 - (n-n)\ 2^{-28} = 1 - 0$. 1 - 0 in the AR yields all ZEROS in the numerical portion (T2-T29) and an END CARRY into T1 to change the sign bit to 0. The END CARRY set term, AC_s, occurred at $T29 \cdot \overline{CE}$ - the end of the two-word recirculation cycle which yielded n bits of ID and MQ shift.

D-12e The AC_s term can terminate TRANSFER by the CL_r gate qualified by $AC_s \cdot T29 \cdot \overline{CE} \cdot DS \cdot S6 \cdot SW$, active during the execution of a SHIFT command (see dwg. 30). This means that the number in AR can control how many bits of shift the contents of ID and MQ will experience.

D-12f The T number of the SHIFT command can also cause termination of TRANSFER by virtue of $T29 \cdot CC$. The T number of a SHIFT command is RELATIVE hence states how many word times TRANSFER can last. If n bits of shift are desired, T should equal 2n.

D-12g The number of bits of shift can be controlled either by $T29 \cdot CC$ or AC_s. Whichever one of these signals precedes the other will control termination of TRANSFER. The order in which these signals arise is a matter of programming. In any event, AR will be incremented if CH = 0. Should the increment AR feature be unwanted a non-zero characteristic may be used.

D-12h Needless to say, the command must be located in an ODD word time and be IMMEDIATE to yield TRANSFER starting with an EVEN word time to accommodate an integral number of two-word recirculation cycles. The S/D bit can contain anything since it controls nothing associated with this command.

SOURCE 27 : NORMALIZE MQ

D-13a Ref. Drawing 36: A number in MQ is normalized when the highest order numerical bit position ($T29 \cdot \overline{CE}$) contains a ONE. (i.e. $1/2 \leqslant n < 1$). During the time of TRANSFER, a NORMALIZE command will cause MQ to precess, shifting its contents to the left one bit position every two word times. The NORMALIZE command must be written as IMMEDIATE and be located at an ODD word time. Its T number is interpreted as relative.

D-13b As in the case of the SHIFT command, AR is incremented every odd word time of TRANSFER if CH = 0; however, AR has no control over TRANSFER.

Incrementing AR serves only to record the number of bit positions of shift experienced by MQ for later program reference.

D-13c The NORMALIZE operation can be terminated by two different terms. One is the T29·CC which is a function of the T number. The other term is a function of the contents of the T29·$\overline{CE}$ position of MQ itself. If a ONE is written in MQ at T29·$\overline{CE}$ time, the PM flip-flop will be set; a ZERO written at that time will reset PM. When a number in MQ is being shifted left, PM will remain reset until a ONE is shifted up to the highest order bit position, hence written at T29·$\overline{CE}$ - setting PM. When PM is set, PM·DS·S6·SX will terminate TRANSFER, stopping the shift. As in the case of the SHIFT command, whichever terminating signal arises first will terminate TRANSFER.

D-13d Since a T29·$\overline{CE}$ pulse sets PM, and PM terminates TRANSFER, TRANSFER will not be terminated until the end of Tl·CE (due to the one pulse period delay between the rise of a. SET term and CLOCK). This is tolerable since during Tl·CE time nothing can be written in MQ to alter its contents because $\overline{TE}$ qualifies the writing term during precession. The advantage in using PM to terminate TRANSFER rather than another suitable pulse (such as PM_S) is that if a number to be NORMALIZED already has a ONE in its T29·$\overline{CE}$ position when TRANSFER starts, TRANSFER will be terminated by PM after one pulse period preventing any shift. The only result of TRANSFER lasting one pulse period will be blocking recirculation of the sign bit which should be ZERO anyway.

SOURCES 28 AND 29: TEST COMMANDS

D-14a These commands all test certain signals. In the event the selected signal is present, CQ will be set and the next command will be read during WT = N+1 instead of WT = N. (Ref. dwg. 30 and sec. C-19). One word time of TRANSFER is all that is required, so these commands are ordinarily written IMMEDIATE with T = L+2.

D-14b SOURCE 28·CH = 0, TEST "READY": DS·S7·SU·④·READY sets CQ. READY is a signal which is high whenever the INPUT-OUTPUT system is idle and is defined as $\overline{OC1}\cdot\overline{OC2}\cdot\overline{OC3}\cdot\overline{OC4}\cdot\overline{OD}$. READY is ordinarily interrogated prior to the initiation of a new INPUT-OUTPUT operation.

D-14c SOURCE 28·CH = 1, TEST "READY IN": DS·S7·SU·①·READY IN sets CQ. READY IN is a signal which can originate in the INPUT-OUTPUT REGISTER accessory.

D-14d SOURCE 28·CH = 2, TEST "READY OUT": DS·S7·SU·②·READY OUT sets CQ. READY OUT can originate in the INPUT-OUTPUT REGISTER.

D-14e SOURCE 28·CH = 3, TEST "DA-1 OFF" DS·S7·SU·③·$\overline{GO}$ sets CQ. $\overline{GO}$ originates in the DA-1 (DIGITAL DIFFERENTIAL ANALYZER) accessory.

D-14f SOURCE 29, TEST OVERFLOW: DS·S7·SV·FO sets CQ. If the OVERFLOW FLIP-FLOP (FO) was set during some previous operation, this command will interrogate it and set CQ accordingly. Upon interrogation, FO is reset by DS·S7·SV·FO·CQ, at T2 time.

SOURCE 30: MAGNETIC TAPE WRITE FILE CODE

D-15 (See sec. F-13)

SOURCE 31·CH = 0: NEXT COMMAND FROM AR

D-16a This command allows the following command (and the following command only) to be read from AR instead of the normal command line established by the configuration in the CD flip-flops. READ COMMAND timing considerations remain unchanged. One word time of TRANSFER is sufficient, hence the command is usually written IMMEDIATE with T = L+2. During TRANSFER, DS·S7·SX·④ sets CG (ref. dwg. 22). When the next command is read, CG permits $\overline{\text{AR}}$·RC to constitute the INVERTED COMMAND and simultaneously blocks MC, which is the normal inverted command line information. After this next command has been read, T29·RC resets CG and normal operation resumes.

D-16b (<F>·<SA> + TAPE START)·WORD 107 + <OP> is applied to the reset term of CG so that AR will not be the command source for the first command executed after the turn-on cycle or after use of the F or P key functions. (The <OP> term is a redundancy since TAPE START includes AUTO TAPE START, which rises during the turn-on cycle. For this reason, the <OP> term has been eliminated from CG_r in some models.)

SOURCE 31·CH = 1: TRANSFER NT to LINE 18

D-17 The NUMBER TRACK (CN) contains information for timing purposes; however, this information constitutes a handy source of constants especially useful for diagnostic routines. CN may be transferred to LINE 18 (M18) by the M18 writing term DS·S7·SX·①. No provision is made for blocking recirculation of M18 by this command; therefore, a command clearing the appropriate word(s) of M18 should precede the operation unless M18 is known to be cleared. Otherwise, M18 will contain the logical sum of M18 and the NUMBER TRACK in the words affected by this command.

"DO NOTHING" COMMANDS

D-18 A variety of SOURCE, DESTINATION, and CHARACTERISTIC configurations will cause no activity other than cycling the computer through its states. If the intent is to "kill time", the recommended "do nothing" command is: SOURCE = 00, DESTINATION = 00, CH = 0. Use of other "do nothing" configurations could conflict with future applications of optional peripheral devices.

SECTION E

MANUAL CONTROLS

This section, although small, may constitute the handiest reference material for the reader who will actually operate and/or maintain a G-15D. Many of these controls merely duplicate the functions of commands.

MANUAL OPERATIONS - GENERAL

E-1a The G15 is equipped with several switches to perform logical functions. Most of these are located on the typewriter, though some are located inside the computer on a maintenance panel.

E-1b Drawing 42 shows the typewriter keyboard and the switch panel with indications of the functions. In general, the switches are for the purpose of setting up the computer prior to the performance of a program. They are also useful for diagnostic purposes.

E-1c Many of the controls deal with INPUT-OUTPUT functions or other material covered in sections of this manual to follow. These controls will be mentioned in this section; however, to fully understand their functions it is necessary to understand the associated circuits.

E-1d Many of the manual controls are typewriter keys. Associated with each typewriter key is a mechanical switch which will close for at least 30 milliseconds (one drum cycle) when the key is depressed. When a particular switch closes, its floating point is grounded yielding 0V on its output wire; when the switch is open, its floating point is held at -20V by a PULL-DOWN resistor. For example, if the "M" key is depressed, its associated switch contact closes, rendering the output signal, ⟨M⟩, at 0V (see dwg. 43 - upper right.) If the switch is not closed, a 1k PULL-DOWN resistor to -20V renders the output signal, ⟨M⟩, at -20V. In other words, ⟨M⟩ can qualify a gate if the "M" key is activated.

THE ENABLE SWITCH - ⟨SA⟩

E-2 The ENABLE SWITCH, located on a component chassis underneath the typewriter, is a toggle switch,which, when closed, will yield the signal ⟨SA⟩. ⟨SA⟩, sometimes called "SAFETY", qualifies certain gates in the computer for purposes which will become apparent as the other switches are discussed. In general, when most typewriter keys are depressed, the ENABLE switch must be ON for the computer function to be performed. This prevents accidental activation of a key from affecting the computer. If the notation ⟨M⟩⟨SA⟩ is made, this means that the "M" key is depressed with the ENABLE switch ON. In addition to its gating functions, ⟨SA⟩ can interrupt a TYPE-OUT operation (sec. F-8u.)

THE PUNCH SWITCH - ⟨MAN PUNCH⟩

E-3 This switch, if ON, yields the signal ⟨MAN PUNCH⟩. ⟨MAN PUNCH⟩ will cause the punch to operate during TYPE-OUT operations (sec. F-8ae). The state of this switch may be interrogated by a TEST command (sec. D-36).

THE COMPUTE SWITCH - ⟨$\overline{GO}$⟩, ⟨GO⟩, and ⟨BP⟩

E-4 See section C-20.

"I" KEY - SINGLE CYCLE - <I>

E-5 If the COMPUTE switch is on the <$\overline{\text{GO}}$> position, activating the "I" key (with the ENABLE switch ON) will allow one command to be executed. The sequence of events is as follows (ref. dwg. 30, upper left):

(1) $\overline{\text{CH}}\cdot\overline{\text{CZ}}$: initial condition established by absence of <GO> + <BP>.

(2) $\overline{\text{CH}}\cdot\text{CZ}$: <I>·<SA> allows TO to set CZ. $\overline{\text{CH}}\cdot\text{CZ}$ lets T21·CC set CJ and allow a command to be read (i.e. RC is generated).

(3) CH·CZ: when command is read, RC·<$\overline{\text{GO}}$> sets CH preventing another command from being read.

(4) $\text{CH}\cdot\overline{\text{CZ}}$: when <I> drops to -20V the next TO resets CZ.

(5) $\overline{\text{CH}}\cdot\overline{\text{CZ}}$: $\overline{\text{CZ}}$ resets CH. (This was the initial condition.)

TO is used to control CZ to prevent more than one command from being read as a result of "I" key contact bounce. The SINGLE CYCLE feature should be used with caution when RETURN EXIT commands exist in a program (sec. D-61).

"C" KEY - SET COMMAND LINE = 00 - <C>

E-6 Ref. Drawing 22 - lower right: <C><SA> resets the CD flip-flops yielding the 000 configuration. $\overline{\text{CD1}}\cdot\overline{\text{CD2}}\cdot\overline{\text{CD3}}$ qualifies the gate permitting $\overline{\text{MO}}$ to become $\overline{\text{MC}}$, thereby rendering LINE 00 the COMMAND LINE.

"1" - "7" KEYS - SET COMMAND LINE

E-7a These are dual function keys. They are used primarily for generation of codes to supply digital data to the computer under control of the INPUT-OUTPUT SYSTEM. Their other function is that of controlling the CD flip-flops, hence the COMMAND LINE.

E-7b In the component chassis under the typewriter is an array of diode OR gates known as the TYPEWRITER OUTPUT MATRIX. This is an encoding circuit which yields '5-level' codes as a function of several data-input and input-control keys. (This will be discussed more fully in sec. F-3c-f).

E-7c For the time being, the 1-7 keys generate codes such that signals known as TYPE 1, TYPE 2, and TYPE 3 assume 7 different configurations. These signals, providing <SA> is high, and the INPUT-OUTPUT system is idle, can set the CD flip-flops, hence the command line as follows:

Key	Type 3	Type 2	Type 1	CD3	CD2	CD1	Comm. Line
(C				0	0	0	00)
1			1	0	0	1	01
2		1		0	1	0	02
3		1	1	0	1	1	03
4	1			1	0	0	04
5	1		1	1	0	1	05
6	1	1		1	1	0	19
7	1	1	1	1	1	1	23

E-7d The CD's must be reset prior to each set-up operation since the TYPE levels will only serve to set the CD flip-flops and will not reset them. Ordinarily, to manually set a new command line, the sequence is: <C>·<SA>, then "1"-"7" <SA>. Drawing 22 illustrates the circuit.

"F" KEY - SET "N" = 00 - <F>

E-8a The "N" number in the COMMAND REGISTER causes the generation of T21·CC during WT = N-1 to initiate RC. N is ordinarily a function of the previous command; however, means must be provided to establish the initial N number for the first command in a program.

E-8b The "F" key provides this feature by manipulating the COMMAND REGISTER such that on subsequent cycles a T21·CC will occur during WT 107, causing the first command to be read at WT 00.

E-8c (Ref. dwg. 22 - left) The "F" key will only be used when the computer is idling in the WAIT TO READ COMMAND state. This means that: (1) $\overline{CJ}$ will remain high, and (2) WORD 107 (= T1·CN + CT + TO) will be high during all of word 107 of each drum cycle. When the "F" key is activated, with the ENABLE switch ON, <F>·<SA>·WORD 107·$\overline{CJ}$ will raise the AUGEND term of the COMMAND REGISTER, inserting all "1's" during word 107. As a result of the arithmetic in the COMMAND REGISTER, a T21·CC will be generated during word 107 of all subsequent drum cycles and will result in the next command being read from WORD 00 when the COMPUTE switch is placed on GO or BP.

E-8d It should also be noted that since during word 107 all CM AUGEND bits are rendered ONES, any previously established MARK established by a MARK EXIT command is obliterated and word 00 is substituted. (The effect on the T29·CC, which establishes TRANSFER timing, is of no consequence since it will be altered by the next command which is read.)

E-8e <F>·<SA>·WORD 107 also sets CH and resets CQ and CG, assuring that (1) the computation cycle is stopped, (2) the first command is read from word 00 and not 01, and (3) that the first command is not read from AR but from the line selected by the configuration of the CD flip-flops.

"M" and "R" KEYS - MARK and RETURN - <M> and <R>

E-9a These keys act as a pair. (Do not confuse their functions with the MARK EXIT and RETURN EXIT commands.) Frequently it is desirable to interrupt a problem and manually perform some functions which may serve to alter the contents of the ACCUMULATOR REGISTER (AR) and/or the COMMAND REGISTER (CM). In such cases it is desirable to store the original contents of AR and CM, perform the manipulations, then return the original contents to AR and CM. MARK stores the information, RETURN returns it.

E-9b MARK: (Ref. Dwg. 59) The signal MARK (=<M><SA>·["WORD 107"]) causes the contents of AR to be written into word 107 of LINE 1. Simultaneously $\overline{\text{MARK}}$ blocks recirculation of LINE 1. At the same time, MARK causes the inverted contents of CM (i.e. $\overline{CM}$) to be written into word 107 of LINE 0; $\overline{\text{MARK}}$ blocks recirculation of LINE 0 at this time. This suggests that word 107 of LINES 0 and 1 should be reserved for this function and not employed in any program.

E-9c RETURN: (Ref. Dwg. 59) The signal RETURN (=<R><SA>·["WORD 107"]) blocks recirculation of AR (AUGEND term) during word 107 inserting M1 as the ADDEND.

The writing term will be the original contents of AR prior to the MARK operation. RETURN substitutes $\overline{MO}$ for CM as the AUGEND term of the COMMAND REGISTER during word 107. The double inversion experienced by original CM information is compensatory. Since MARK and RETURN both performed their functions during word 107, the arithmetic in the COMMAND REGISTER is effectively unaltered and T29·CC, T21·CC, and T13·CC will appear during the same word times as they did prior to the MARK.

"T" KEY - "N" to AR - <T>

E-10a The object in this operation is to transfer "N" number information to the AR where it can be viewed on an oscilloscope or typed out. This is particularly useful in "checking out" new programs to see if they follow the sequence anticipated by the programmer.

E-10b Since the "N" number is in the dynamic portion of a command, it is sent to the COMMAND REGISTER and modified every word time by the arithmetic, presenting a monitoring problem. Therefore, evidence of the "N" number, T21·CC, is used to obtain a constant representing "N". The NUMBER TRACK (CN) contents immediately following T21·CC consists of the 'WT+1' factor. Since T21·CC occurs during WT = N-1, and the contents of CN during that word time equals WT+1, the CN data = WT+1 = (N-1) +1 = N. This data, during T22-T28, is transferred from CN to AR as illustrated on Drawing 59. The T1-T21 portion of AR remains unchanged.

E-10c When N = 00, T21·CC occurs during WT 107 yielding CN information =20 rather that WT+1 (See Dwg. 6). This results in ambiguity between the indications from CN arising from N = 00 and N = 20. For this reason, whenever T21·CC occurs during word 107, TO is written into AR yielding 10010100 as the configuration in the highest order eight bits of AR. (i.e. $94_{16} \equiv 00_{10}$.)

E-10d Usually when N is transferred to AR by "T" key operation, an AR TYPE-OUT follows (initiated by the "A" key). AR is ordinarily typed out in terms of SIGN and 7 HEXADECIMAL DIGITS. The highest order two hex. digits indicate the "N" number (i.e. LOCATION of the next command). The table on Dwg. 4 can be used for hex. to decimal conversion.

E-10e Needless to say, the original contents of AR are disturbed, therefore the MARK and RETURN keys could be used to advantage. A typical sequence in SINGLE CYCLING a new program and producing a record of command locations is: M-T-A-R-I-M-T-A-R-I-M-T-A-R-I-----.

"A", "Q", "P", "B", and "S" KEYS - INPUT/OUTPUT

E-11a These keys are for purposes of manually initiating or terminating INPUT-OUTPUT Operations. Reference is directed to the INPUT-OUTPUT section of this manual (sec. F) and to Drawing 45.

E-11b "A" KEY - TYPE AR: <A><SA>·Ⓝ sets the OC's to a 1000 configuration and sets the OD flip-flop, initiating a TYPE-OUT AR. <SA> must be released before any typing will occur.

E-11c "Q" KEY - SET TO TYPE IN: ⟨Q⟩⟨SA⟩ sets the OC's to a 1100 configuration, rendering the INPUT system receptive to typewriter input-data and input-control keys.

E-11d "P" KEY - PHOTO-TAPE READ, SET COMMAND LINE = 23 and N = 00: ⟨P⟩⟨SA⟩ sets the OC's to a 1111 configuration, starting the PHOTO-TAPE READER. Also, ⟨P⟩⟨SA⟩ generates the TAPE START signal (ref. dwg. 22). TAPE START·["WORD 107"] duplicates the function of the "F" KEY, setting N = 00. TAPE START also sets the CD's to a 111 configuration, establishing LINE 23 as the COMMAND LINE. The last two functions are a convenience for the operator.

E-11e "B" KEY - START PHOTO-TAPE REVERSE CYCLE: ⟨B⟩⟨SA⟩ sets the OC's to a 0110 configuration, initiating a PHOTO-TAPE REVERSE CYCLE (see sec. F-5)

E-11f "S" KEY - STOP INPUT-OUTPUT: $\langle S\rangle\cdot(\langle SA\rangle+\overline{OC1}\cdot\overline{OC2})$ resets the OC flip-flops, manually terminating any INPUT-OUTPUT operation. It is not necessary to hold the ENABLE switch ON when using the "S" KEY to terminate a TYPE-IN. (Do not confuse the function of this key with that of the STOP code.)

LOGIC PUSH-BUTTON MAINTENANCE SWITCHES

E-12a A small maintenance panel is located inside the computer behind the power control panel. This panel includes many banana jacks yielding frequently needed oscilloscope signals for "SYNC." and "Y-AXIS". Also, six push-button switches exist as follows:

CLEAR	SET
M19	M19
M23	OP
NT	NT

E-12b These switches perform logical functions associated with the initial TURN-ON CYCLE and are described in the section on that subject - Section G-3.

SECTION F

THE INPUT-OUTPUT SYSTEM

This section accounts for the subdivision of the computer without which there could be no communication between the computer and the operator. Prerequisite knowledge should include the contents of Sections A, B, and C-15. The principle of PRECESSION should also be understood; examples of this are given in Sections D-10c-j.

THE INPUT-OUTPUT SYSTEM - GENERAL

F-1a The INPUT-OUTPUT SYSTEM of the G-15D may almost be considered an independent data processing system. Its logical linkage to the COMPUTER PROPER is very loose. It utilizes some of the basic timing signals and two of the programmable memory lines, M19 and M23. Another memory line, MZ (a 4 word line), is "private property" of the INPUT-OUTPUT system and facilitates certain data handling functions. An INPUT or OUTPUT operation and a program can progress concurrently as long as LINES 19 and 23 are not involved in the program.

F-1b The different INPUT-OUTPUT operations available are listed on Drawing 3 as SOURCES 00-15 and 30. The SOURCE 30 operation is a special case and does not employ the INPUT-OUTPUT SYSTEM. It will be discussed separately in sec. F-13.

F-1c Note that the sixteen INPUT-OUTPUT operations (defined by SOURCES 00-15) are divided into four categories: FAST-OUT, FAST-IN, SLOW-OUT, and SLOW-IN. These states of the INPUT-OUTPUT system define the nature of the operation in progress; further subdivision selects the particular INPUT-OUTPUT medium to be active.

F-1d A central control circuit for all INPUT-OUTPUT operations exists in the form of a 4-bit static register composed of the OC FLIP-FLOPS: OC1, OC2, OC3, and OC4. Note on Drawing 3 that each operation is represented by a binary configuration in these flip-flops. The configurations are the same as the SOURCE number (i.e., 00-15).

F-1e In general, when no INPUT-OUTPUT operation is in process the OC configuration is 0000 - representing the "READY" state. A particular INPUT-OUTPUT operation may be initiated by setting these flip-flops to the appropriate configuration. While they are set in a non-zero configuration the INPUT or OUTPUT operation takes place. Normally a signal within the INPUT-OUTPUT SYSTEM itself will call for termination of the operation and will accomplish this end by resetting the OC flip-flops to the all-ZERO configuration (READY state). The latter may also be accomplished manually ("S" Key), or by a special command (SET READY). Once the READY state is established, a new INPUT-OUTPUT operation may be initiated. (The special command, "TEST READY", permits a programmable interrogation of this state - ref. sec. D-14b.)

F-1f The contents of the OC flip-flops are decoded, yielding signals which serve to control the logic of the INPUT-OUTPUT circuits and also the activity of the various electro-mechanical INPUT-OUTPUT media. The means of controlling the OC's varies. A glance at Drawing 45 should acquaint the reader with the various OC control options and the output signals from the decoding gates. Neon indicators on the front panel monitor the state of the OC'S at all times.

OC FLIP-FLOP CONTROL

F-2a PROGRAMMED OC SET-UP: Whenever the READY state is present, an INPUT-OUTPUT operation may be initiated by a special command. Such a command involves DESTINATION - 31 and SOURCE = 00 - 15. In these cases the contents of the SOURCE flip-flops in the CONTROL SWITCH will contain a configuration such that the 2^4 bit is equal to ZERO. The 2^4 static flip-flop is CV; therefore, during the time of TRANSFER, $DS \cdot \overline{CV}$ will be high and is used to transfer the 2^0 - 2^3 source information to the OC'S as shown at the top of Drawing 45. Once this OC configuration is established, new commands may be read and executed independently of the INPUT-OUTPUT system, provided LINES 19 and 23 are not involved in the program.

F-2b MANUAL OC SET-UP: This has been partially covered in sec. E-11. Drawing 45 (left) illustrates the options. More will be said about these functions under the appropriate headings (e.g., the "A" key, hence <A><SA>, will be discussed in sec. F-8, which includes AR TYPE-OUT).

F-2c OC RESET: Drawing 45 (lower left) shows these options. As the notes suggest, the means of resetting the OC'S (other than manually) depend upon the nature of their configuration. These too will be discussed under the appropriate headings.

THE INPUT MEDIA

F-3a When the OC'S assume the configuration 11xx, a SLOW IN operation is called for. The highest order two bits (configuration OC4·OC3) set up the logic circuits with qualifying signals IN and SLOW IN. The lowest order two bits determine which input medium is to be active but do not in all cases directly control the particular medium.

F-3b Before the behavior of the INPUT LOGIC circuits can be intelligently discussed, the INPUT MEDIA themselves must be understood. The sections to follow will attempt to accomplish this end.

F-3c TYPEWRITER: Ref. dwg. 43) when the typewriter is to be used as the input medium, manual activation of certain keys supplies data-input characters (hexadecimal) and functions in terms of 5-level codes. These codes appear on five signal wires: TYPE 1, TYPE 2, TYPE 3, TYPE 4, and TYPE 5. Drawing 46 in the INPUT-ORIGIN-TYPEWRITER column indicates the active keys and the codes they generate.

F-3d Generation of these 5-level codes is accomplished in an encoding circuit known as the TYPEWRITER OUTPUT MATRIX. This is nothing more than an array of OR gates such that when any of the 20 typewriter keys which produce codes is activated, its switch contacts will raise the appropriate output level lines to OV to represent ONES (see dwg. 43). Any level line not raised to OV by an OR gate diode is held at -20V by a 4.7k pull-down resistor (indicated on the DETAILED PRINT - 3D282). Each one of the five output level lines except Type 4 is provided with an R-C filter to integrate the signal with a suitable time constant to reduce signal irregularities due to point bounce and transient effects. This is necessary to prevent an electronic synchronizing circuit in the computer from being illegally triggered. These output level lines contain the typewriter output signals for the INPUT system.

F-3e In addition to these level signals, "F-B" reaches the computer for synchronization purposes. This signal is required for proper operation of the INPUT-OUTPUT system when the typewriter is used as the OUTPUT medium; however, it will reach the input system and must be mentioned. The signal will be high if the SPACE BAR, SPACE KEY or PERIOD key is activated. The line is provided with an R-C filter since this signal will be used to trigger the synchronization circuit mentioned above. This function will have no effect on the INPUT system as far as data entry is concerned.

F-3f All of the signals mentioned above may be expected to last at least 30 milliseconds (over one DRUM CYCLE). 1/10 second is the average duration.

F-3g PHOTO-TAPE READER: (Réf. dwg. 44) This device "reads" the 5-level codes punched on a paper tape. The nature and sequence of the codes are a function of how the tape to be read was punched. This is ordinarily a function of the OUTPUT system of the computer and the programming thereof. The PHOTO-TAPE READER supplies the INPUT SYSTEM with the codes on 5 level lines: PHOTO 1, PHOTO 2, PHOTO 3, PHOTO 4 and PHOTO 5. These signals are similar to those originating in the typewriter but are of much shorter duration and occur in rapid succession. The PHOTO-TAPE READER will supply, as output, any code appearing on the tape being read; however, only the codes indicated on Drawing 46, column INPUT-ORIGIN-PAPER TAPE should ever appear.

F-3h Assuming that photo-electric reading circuits exist, the problem of tape motion will now be considered. Motion of the tape can assume two directions past the reading station - FORWARD and REVERSE. The direction is a function of torque delivered by two motors which constitute the prime-movers of reels containing tape in a magazine. (Which reel is the FEEDER and which is the COLLECTOR varies with the direction of motion.)

F-3i Torque applied to the reels is a function of two relays: the FORWARD relay (RY-A) and the REVERSE relay (RY-B). Assume that the FORWARD relay is energized. Points on this relay will accomplish the following:

1) Apply 115 VAC to the upper winding of the two-phase TOP MOTOR.

2) Apply 115 VAC to the lower winding of the TOP MOTOR via the 1.2mf starting condenser C1 such that the phase relationship results in counterclockwise (from front) torque.

3) Apply reduced AC (function of R12 setting) to the upper winding of the BOTTOM MOTOR.

4) Apply reduced AC to the lower winding of the BOTTOM MOTOR via the 1.2mf starting condenser C2 such that phase relationship results in counterclockwise (from front) torque. The amount of torque is sufficient to overcome friction of the feed reel system.

5) Complete circuit to energize illuminator lamp (for photo-electric cirruit).

6) Charge C3A and C3B (80 mfd each) to peak value of line voltage to prepare for "dynamic braking".

F-3j The result of the above is that the tape will move in a FORWARD (upward) direction past the reading station and will be illuminated. The linear speed of the tape is a function of the angular velocity of the motors and the radius to the point of tangency of the tape on the reels. Therefore, no specific linear speed can be given. Linear speed is considered to average 20 inches/sec yielding an average rate of 200 characters/sec. Applying assisting torque (reduced) to the feed reel overcomes friction and improves acceleration characteristics. (Too much assisting torque could result in tape spillage. A 1k potentiometer, R12, is adjusted so that with no magazine on the reader, the feed reel drive is just on the verge of creeping.)

F-3k Had the REVERSE relay been energized instead of the FORWARD relay, the above functions would have been performed except that the torques would have been clockwise and the BOTTOM MOTOR would have yielded full torque while the TOP MOTOR would have yielded reduced (assisting) torque.

F-3l Whenever either relay is de-energized yielding the "no-go" state following a "go" state, the charged 80 mfd condensers (C3A and C3B) are connected by relay points to the windings of each motor supplying "dynamic braking". By the time the charges have been spent, the motors should be stopped and the circuit static. Various R-C and L-C circuits distributed about the schematic are for transient suppression purposes.

F-3m Relay control: Controlling the relays controls the reader. Usually the relays are energized by puller tubes caused to conduct by signals from the OC flip-flops (FORWARD = 1111, REVERSE = 0110 or 0111). The puller tubes may also be controlled by a front panel switch: FORWARD-REWIND. It turns on the appropriate puller tube but breaks the illuminator lamp circuit to prevent any reading activity.

F-3n Reading activity assumes that the tape is in motion and the illuminator lamp is lit. There are six channels to be read: LEVELS 1-5 and SPROCKET HOLE. The SPROCKET HOLE is read to provide a time-gating signal to probe for the presence or absence of LEVELS 1-5. The gating accomplishes the following: (1) it assures that the output signals on the level lines are simultaneous and of roughly the same shape, and (2) it overcomes the adjacent-hole problem, in which the signal between adjacent holes does not decay to zero signal level due to the optics. This is illustrated at the bottom of Drawing 44.

F-3o IN77B photo-diodes are used as the photo-sensitive elements for each channel. These diodes have the following characteristics: high back-resistance when dark and low back-resistance when illuminated. By placing such a photo-diode in a voltage-divider such that its back-resistance constitutes a variable resistance in the voltage-divider, a varying potential may be obtained which is a function of the light striking the cell.

F-3p The SPROCKET cell, CR 13, is connected in such an arrangement, with the negative return at -24.6V (jct. R55-R66) and the positive return at +100V. The junction of CR13 and R17 (clamped by CR7 so it cannot exceed 0V) yields an output potential which should equal 0V when the cell is dark and approach -24.6 when the cell is illuminated. Potentiometer R18 (2.5 meg) permits adjustment of the series resistance to "custom-tailor" the voltage divider parameters to the back-resistance characteristics of the particular cell.

F-3q Potentiometer R18 ("S") is available on the side of the PHOTO-READER and is adjusted for a desirable oscilloscope indication at test point "S", which contains the output of a cathode-follower (V2A, pin 3). This follows the potential at the CR13-R17 junction. Desirable indication consists of maximum signal output (approaching -20V) with no base-line (OV) noise. If the adjustment renders the circuit too sensitive to light, translucency of tape can be a problem.

F-3r The SPROCKET level cathode-follower (V2A) supplies a gating signal to the five other channels such that only during the time of the SPROCKET signals can the other levels yield an output signal if illuminated. The five voltage dividers in the other channels resemble that of SPROCKET except that the negative returns are connected to the SPROCKET channel cathode-follower. This means that between sprocket holes, when the cathode-follower output is at OV, the level photo-diodes (cathodes) cannot yield a signal less than OV regardless of illumination. However, should a level be illuminated, during the time that the sprocket signal approaches -20V, the level photo-diode will yield a signal approaching -20V.

F-3s Such a level signal is inverted by its associated triode. The plate circuit of such a triode yields a positive-going signal clamped at OV and with a base line of approximately -20V (negative clamping is not necessary considering the nature of the circuits to follow). A degenerative circuit is connected to the cathode of each triode. This will not alter the peak-to-peak value of the output signals but will offer reduced gain to undesirable high frequency components on the leading and trailing edges. These high frequency components are further suppressed by .0018 mfd integrating condensers on the output lines. This reduces the possibility of transient-triggering of the synchronization circuit in the INPUT system of the computer.

F-3t Adjustments are available on the side of the reader to adjust the response of each channel. Associated with each adjustment is a test point. The adjustments are made in a manner similar to that of sprocket. Periodic observation of these test points is advisable.

F-3u MAGNETIC UNITS: These supply codes on five level lines in the form of square waves (with peaks of -20V and OV) which appear at a peak rate of 463 characters/sec (i.e. 1 code every 8 word times). The signals will appear whenever the tape transport mechanism is programmed to be in motion and the tape has data recorded upon it.

F-3v CARD UNIT: (IBM 026 with adapter circuits) Mechanical contacts supply level signals in a manner similar to that in the typewriter. In addition to the 5 level signals, a MINUS SIGN signal can be generated and will enter the computer INPUT system independently of the levels.

SLOW-IN LOGIC

F-4a The INPUT SYSTEM will respond to codes from the input media, e.g. the typewriter, photo-tape reader, and magnetic tape units. The codes to which the INPUT SYSTEM will respond are indicated on page 46. Note that all codes for which level 5 is a one are considered numerical hexadecimal characters. When level 5 is a zero the code indicates a control function. The SLOW-IN logical hardware is shown on page 48. An illustrative SLOW-IN timing example is given on page 49.

F-4b Any code arriving at the input levels (ref. dwg. 48 - bottom) will cause the HC buffer-inverter to become high. This will activate a synchronizing circuit which will control the subsequent activity of the input system. Fourth input level control of HC is absent, but note that when fourth level does occur it will always be accompanied by fifth level. In this case fifth level will provide the necessary synchronizing signal. The synchronization circuit is composed of HC, OF1, OF2, and associated gates.

F-4c The signal HC, which triggers the synchronization circuit, must be reasonably free of transients to avoid multiple triggering. This was handled to some extent in the sources of the level signals. Additional high frequency suppression is accomplished by modifying the BUFFER-INVERTER package itself which includes the HC circuit (see dwg. 13). It is essential that this modified package be plugged into the appropriate jack. The package can be identified by a notation on its handle.

F-4d Note that the gates associated with the OF flip-flops in the synchronization circuit are all qualified by "IN". This arises from certain configurations of the OC'S, all of which represent INPUT operations. The OF flip-flops are used for different purposes during OUTPUT operations and are controlled by different gates. In such cases the signal "IN" is low, and the gates it qualifies are effectively out of the circuit. This principle of multiple use of tube circuits is repeated throughout the INPUT-OUTPUT system and accounts for component economy.

F-4e Activation of HC by an input code will set OF1. OF1·IN will enable the input code at the input levels to be inserted into a 5-bit static buffer register consisting of the OB flip-flops. Once the code is in the OB's the signals on the level lines can decay.

F-4f OF1 also sets OF2. HC will decay upon the decay of the input level signals, and the synchronizing circuit will perform its closing cycle yielding the signal Ⓔ, which controls the activity in the INPUT logic circuits.

F-4g Ⓔ is a gated TF pulse. TF is a basic G15D timing signal. $TF=T29\cdot\overline{CE}\cdot CF$, therefore occurs during WORD TIMES CONGRUENT TO 3 MODULO 4, i.e. at T29 of words 3, 7, 11...103, 107. TF is suitable for initiating or terminating activities dealing with 4-word lines at the beginning or the end of their four-word recirculation cycles (ref. sec. C-15). Ⓔ will occur at the first TF pulse after OF1 is reset which will be just after the input levels decay. The TF pulse which is gated to become a Ⓔ is also used to reset OF2, thereby freeing the synchronizing circuit to generate a new trigger from the next input code. A Ⓔ will occur during any INPUT operation after a code is inserted into the OB's. Ⓢ = Ⓔ·OC4 = Ⓔ during SLOW-IN operations only.

F-4h If prior to generation of the Ⓔ associated with one code, a new code arrives at the level lines, the OF2 term disqualifying the OF1 set gate inhibits the next synchronization cycle until Ⓔ is generated. This prevents skipping codes due to loss of Ⓔ at high input rates.

F-4i Once Ⓔ is generated by the synchronization circuit, it will start the logic circuits functioning under control of the code in the OB'S. There is a decoder yielding several signals defining the code residing in the OB'S. (Like many circuits requiring no clarification, the OB decoding gates are not shown in the simplified drawings. They may be easily located in the more detailed prints--in this case dwg. # 3D287.)

F-4j A principle employed in most INPUT-OUTPUT operations is PRECESSION. One bit precessions have been used in MULTIPLY, DIVIDE, SHIFT and NORMALIZE. The VIA AR characteristic is a one-word precession. The INPUT SYSTEM involves 1-bit precessions, 4-bit precessions, and 4-word precessions for information positioning purposes. As the reader may recall, a precession to the left involves inserting a delay in the recirculation path of a memory line.

F-4k Assume the case of a hexadecimal character arriving at the input levels (level 5 must be high). The character will be inserted into the OB's (1xxxx) and a Ⓔ will be generated. Because OB5 is high, Ⓔ will allow the contents of OB1-OB4 to be copied into OA4-OA1 respectively. The OB's will also reset at Ⓔ time freeing them to receive a subsequent code. Ⓢ (= Ⓔ·SLOW) will reset OF3 and set OG because the code in the OB's defines a digit (1xxxx). OG will be high for four word times (terminated at the next TF pulse) and the following will occur:

1) Normal recirculation of M23 will be disqualified - because $\overline{OG}$ is low.

2) OG and $\overline{OF3}$ will enable a precession path for M23 such that M23's read flip-flop will feed OA1. OA1 will feed OA2, etc., and OA4 will feed a gate enabled by $\overline{OF3}$ which will write into M23.

3) During the first four pulse periods the code in the OA's will be written into M23 (this was the original new input hexadecimal character) and....

4) The original contents of M23 will be delayed 4 bit times (or precessed to the left) and hence will be rewritten 4 bit times late.

5) This path is enabled for four word times. At the end of the four word times normal recirculation of M23 is resumed.

6) The following changes will now be noted in M23's contents. (1) The lowest order four bits (T1-T4 of word 00) contain the newest input code, (2) the original M23 information has been shifted four pulse periods to the left, and (3) the original highest order four bits of M23 have been discarded.

7) Each hexadecimal input code will have the effect of entering four bits into the lowest four bit positions of M23.

F-4l Commands or numbers may be entered into the G15D by first converting them to hexadecimal. The procedure to accomplish the conversion of commands into hexadecimal is outlined on drawing 5. The procedure for converting decimal numbers to hexadecimal may be accomplished by simple arithmetic or by consulting a conversion table (drawing 4). To enter a word into the

G15D (be it a number or command) the following sequences of codes could occur:

(SIGN) - 7 HEX. DIGITS - TAB or CARRIAGE RETURN

The total effect of the above will be to enter 29 bits of information into M23 word 00. It may be noted that the sign may be entered at any time providing it precedes the TAB or CARRIAGE RETURN code. The reason for this flexability will be explained later in this section. In practice the programmer uses a program preparation routine which allows him to communicate with the computer using binary-coded decimal leaving the drudgery of conversion to the computer.

F-4m Should a minus sign code (00001) appear at the input levels the following sequence of events will occur:

1) The sign code will be inserted into the OB's and a Ⓔ will be generated as was the case for a hex. code in section F-4k.

2) With $[\text{SIGN}]_{OB}$, 0x001, at Ⓔ time OS is set.

3) Should it have been desired to enter a plus sign no code entry would have been necessary as OS would have been reset already by a previous TAB or CARRIAGE RETURN code or the READY signal.

4) Note entry of the minus sign code has only succeeded in setting the OS flip-flop and not actually entering this information into G15D memory. Further processing of the sign will be accomplished upon receipt of a TAB or CARRIAGE RETURN at the input levels. OS will remember the sign until then.

F-4n Entry of the hexadecimal portion of the word procedes character by character as described in section F-4k. Normally 7 hex. characters will be entered thus these bits will be located in T1-T28 of word 00 of M23.

F-4o The sign bit, presently residing in the OS flip-flop, represents the final bit of data to be entered into M23. Its position should be T1 of word 00. A TAB code (00011) or CARRIAGE RETURN code (00010) entry will give rise to the following sequence of events:

1) The TAB or CARRIAGE RETURN code will be inserted into the OB's and a Ⓔ will be generated as in the case for the hex. character in section F-4k.

2) With $[\text{TAB + CARRIAGE RETURN}]_{OB}$, 0x01x, at Ⓔ time the contents of the OS flip-flop will be transferred to OA1, at the same time OS will be reset (hence the sign will be assumed positive for the next word, thus eliminating necessity for ever having a + code).

3) With $[\text{TAB + CARRIAGE RETURN}]_{OB}$, Ⓢ will set OF3 and OG. OG being high for four word times will block M23's normal recirculation path, while OF3 and OG establish a one bit precession path for four word times from M23's read flip-flop to OA1 to a M23 write gate.

4) During the first pulse period the contents of OA1 (the previous contents of OS) will be written into the T1 position of word 00 and all the original contents of M23 will be written one pulse period late.

5) Precession will last four word times. At the end of this time M23's normal recirculation is resumed.

6) The following changes will now be noted in M23's contents. The T1 bit position of word 00 contains the sign information, and the original contents of M23 has been shifted to the left one bit. The highest order bit of M23 (T29 of word 03) has been lost.

7) If previous to this operation 7 hex. characters had been entered then they are all sitting in M23 in word 00 in the desired bit positions.

8) In effect it is the TAB or CARRIAGE RETURN which enters the sign information into the G15D memory.

F-4p Should the above procedure of entering (SIGN), 7 HEX. DIGITS, TAB or CARRIAGE RETURN be repeated, then the first word entered into word 00 would be shifted into word 01 and the most recent entry would now reside in word 00. A maximum of four words may be entered into M23 in this manner. Additional entry would result in the loss of the original word occupying word 03 as it would be shifted out the high end of M23. Therefore a means is provided to store M23 in a long line, M19. The RELOAD code will initiate this function.

F-4q When RELOAD, 00101, arrives in the OB's, the following sequence of activity occurs:

1) Ⓢ will set OD and OY. This will block normal recirculation of a four word line, MZ. OY will stay high four word times causing M23 to be copied into MZ. This frees LINE 23 for reception of new data.

2) At the first T0 after OD was set, OE will set. OE will remain high for one drum cycle. With OD and OE high normal recirculation of MZ and LINE M19 is interrupted, but a four word precession path is established for 108 word times involving MZ and M19. The path is such that beginning at word 00, MZ's contents are written into M19, and M19's contents are written into MZ. M19's original contents reappear at MZ's read flip-flop delayed four word times and are rewritten into M19. After 108 word times normal recirculation of MZ and M19 is resumed.

3) As a result of the precession the original contents of MZ (which came from M23) reside in words 0-3 of line M19. The original contents of M19 are shifted to the left four words. The original highest order four words (104-107) of M19 reside in MZ where they will be lost upon receipt of the next RELOAD code.

F-4r If 27 four-word groups are entered in this manner, LINE 19 will be "filled", with the first word entered residing in word 107. At this time the INPUT operation should be terminated. If the typewriter is the input medium, the "S" key should be activated following the final RELOAD code. This resets the OC flip-flops terminating the INPUT operation. The ENABLE switch need not be ON for this operation since <S>·OC1·OC2 resets the OC's.

F-4s If any other input medium is used it should substitute a STOP code for the final RELOAD code. A STOP code performs not only the OC reset function but also the RELOAD function. When a STOP code resets the OC'S it does so approximately one drum cycle prior to the time that the LINE 19 - MZ precession is completed. This is for the purpose of stopping the PHOTO-READER at the earliest possible time. For this reason the precession gates must be qualified by "FAST $\overline{\text{OUT}}$" as well as "IN" ("FAST $\overline{\text{OUT}}$" includes the 0000 configuration). Also $\overline{\text{OD}}$ must qualify the gate which forms the READY signal so the INPUT-OUTPUT system will not appear to be idle while it is still occupied in this final precession.

F-4t Remaining aspects of the INPUT system should be evident considering the material on Drawings 48 and 49. Miscellaneous elusive facts are as follows:

1) OB5 qualifies the OA1_r gate (Ⓔ$\cdot\overline{\text{OB4}}\cdot$OB5) to prevent the fourth-level ZERO in a TAB or CARRIAGE RETURN code from resetting OA1 at the same time a ONE in OS calls for setting OA1.

2) When a MINUS SIGN is to be stored in OS, the MINUS SIGN code is entered; entry of a PLUS SIGN involves merely the failure to enter a MINUS SIGN since OS is reset every time a TAB or CARRIAGE RETURN is entered. CA-1 CARD EQUIPMENT can set OS without the formality of HC, etc.

3) A buffer-inverter is used to set OG and prevent the possibility of simultaneous reset. This is to permit a high data entry rate involving consecutive TF pulses becoming Ⓔ's.

4) $\overline{\text{TF}}$ qualifies Ⓞ, which permits the contents of the OA's to shift during precessions. Preventing OA shift at TF time has no effect on the desired precession results in LINE 23; however, it does permit a new character to be transferred from OB's to OA's without conflict in the event of consecutive TF pulses becoming Ⓔ's.

5) The purpose in resetting the OC's approximately one drum cycle prior to the completion of the STOP code operation is to stop the input medium with minimum delay. (Once a STOP code reaches the OB's it will remain there until the OC's are reset rendering the INPUT system insensitive to any incoming level codes which may follow.)

6) The SPACE and PERIOD keys on the typewriter generate no codes but do raise HC. This is for the benefit of the typewriter when used as an output medium. A PERIOD code coming from tape will raise HC and be inserted in the OB's; however, no circuit beyond the OB's will respond to it. An incoming SPACE code is no code at all. SPACE is an output function.

7) A WAIT code performs the same data entry function as HEXADECIMAL ZERO. WAIT is an editing feature used primarily for output purposes.

F-4u The codes may arrive at the input at any rate less than 926 characters/sec. (i.e., 1 character/4 WT), except in the case of the RELOAD code which cannot exceed a rate of 17.2/sec. (i.e., 1 character/2 drum cycles). Considering the normal sequence of codes involved in data entry, the RELOAD rate is the limiting factor. In practice, the maximum input rate delivered by any of the media is 463 characters/sec. and 15 RELOADS/sec.

F- 4v It is suggested that the reader study the example on Drawing 49 confirming the activity on Drawing 48. This combines the time axis with the logical hardware-- both of which should be fully understood. (Drawing of colored vertical lines joining key TF dots at the top and bottom of the timing diagram is advisable.)

PHOTO-TAPE SEARCH (FAST-IN)

F-5a FAST-IN involves searching for certain codes either on PAPER or MAGNETIC TAPE with the purpose of positioning tape but not obtaining data. This discussion will be devoted to PHOTO-TAPE search operations only. MAGNETIC TAPE search operations will be discussed in sec. F-14.

F-5b Data ordinarily is punched on tape in the form of 108 word blocks (27 four-word groups) terminated by a STOP code. A few inches of blank tape should be provided between adjacent blocks to allow for tape acceleration and deceleration. When a tape in the PHOTO-READER is read for data entry purposes (i.e., SLOW-IN), it should be started from a blank tape area. The PHOTO-READER will read tape until a STOP code is read; the STOP code causes the OC's to be reset stopping the mechanism.

F-5c Assume now that a block of tape has been read and the tape is stationary in a blank tape area. The programmer wants to back the tape up one block so that the same block can be re-read. The PHOTO-TAPE reverse cycle will accomplish this. (Two PHOTO-TAPE REVERSE CYCLES would back the tape up two blocks.)

F-5d (Ref. dwg. 45) The cycle can be initiated by setting the OC's in the 0110 configuration, representing PHOTO-TAPE REVERSE-PHASE 1. This can be accomplished manually ("B" key) or by a special command (S = 6, D = 31). The configuration yields the signals "IN" and "PHOTO TAPE REVERSE".

F-5e "PHOTO-TAPE REVERSE" energizes the REVERSE relay in the PHOTO-READER, lighting the illuminator lamp and moving the tape in a REVERSE direction. "IN" qualifies the "front end" of the INPUT system (dwg. 48 - bottom) allowing it to register all incoming codes in the OB's and to generate Ⓔ's, however, none of the data goes into any memory lines since "SLOW-IN" is not present.

F-5f (Ref. dwg. 45 - left side) When the first STOP code is registered in the OB's, the signal PHOTO-TAPE REVERSE·[STOP]$_{OB}$·Ⓔ sets OC1 (gate 1), producing the 0111 configuration in the OC's, representing PHOTO-TAPE REVERSE-PHASE 2. The tape continues reading in REVERSE until it encounters the next STOP code; then PHOTO-TAPE REVERSE·OC1·[STOP]$_{OB}$·Ⓔ (gates 1 and 2) sets OC4, yielding the 1111 configuration in the OC's, starting a regular FORWARD PHOTO-TAPE READ operation. This reverses tape motion.

F-5g The FORWARD operation reads tape until the first STOP code is reached. This resets the OC's, stopping the tape (according to normal procedure in SLOW-IN). The complete operation succeeds in backing the tape up one block as illustrated on Drawing 45 (upper right).

F-5h Miscellaneous facts: (1) Among the terms which form Ⓚ (dwg. 48) to reset the OB's is Ⓔ·FAST IN. This permits the OB's to be cleared after the STOP codes have been interrogated during the REVERSE search, permitting normal operation of the $OF1_S$ gate on subsequent codes. (2) During the final part of the cycle, the FORWARD reading will affect LINES 19 and 23 since codes will be read during a SLOW-IN situation. Therefore it is advisable that the contents of these lines be insignificant when the cycle is initiated. (3) Programming a PHOTO-TAPE REVERSE-PHASE 2 may appear worthless, but it can serve the purpose of "homing in" on a particular STOP code when necessary.

SLOW-OUT - GENERAL

F-6a The SLOW-OUT system may almost be considered an independent computer. It is independently programmable to provide maximum flexibility in editing computer output. The destination of information extracted from the selected output memory line (M19 or AR) is a 5-bit buffer register consisting of the OB flip-flops. (The reader may recall that the OB's served as a 5-bit buffer register for incoming codes during INPUT operations.)

F-6b The means of loading the OB's with information is a function of the FORMAT, which is actually a program consisting of a series of instructions known as FORMAT CHARACTERS. The FORMAT CHARACTERS are 3-bit commands and are interrogated and obeyed in a fixed sequence.

F-6c Assuming that the desired output codes appear in the OB's, one of the output media will respond to them. Which output medium will respond to the codes is a function of the particular SLOW-OUT operation selected. As Drawing 3 indicates, the TYPEWRITER, the paper tape PUNCH, and the CA-1 CARD EQUIPMENT are the different options. The OC flip-flops determine which medium will receive an "execute" pulse and respond to the code in the OB's. The PUNCH switch on the front of the TYPEWRITER enables the PUNCH to operate whenever the TYPEWRITER is operated as an output medium.

THE SLOW OUTPUT MEDIA

F-7a TYPEWRITER: The TYPEWRITER has already been mentioned as an INPUT device. In this case activation of the keys was manual. When the TYPEWRITER serves as an OUTPUT medium, the keys are electrically activated by means of solenoids.

F-7b Drawing 43 illustrates the 21 solenoids which can be energized by computer control. These include data and control functions. Which one of the 21 solenoids will be energized by the computer is a function of the contents of the OB flip-flops.

F-7c All solenoids are returned to +160V. Relays RY1-RY5 follow the contents of the OB's. The relay points, in accordance with the configuration of the relays, will complete a path from the points of RY6 either to one or none of the solenoids. If and when RY6 is energized by an "execute" pulse, its points will apply ground to the input of the path just described and energize one solenoid or none at all depending upon the contents of the OB's. Obviously, a code in the OB's representing a given function will result in the activation of the corresponding key (i.e., $10011_{OB} = 3$, therefore the "3" key is pulsed).

F-7d The same codes that appear at INPUT can be delivered to OUTPUT. There is no purpose in the typewriter responding to three of these codes, therefore no accommodations are made for activation of any solenoids in these cases. The codes are: WAIT, RELOAD, and STOP. (Ref. dwg. 46)

F-7e Whenever any key is operated, the associated switch contacts close. This raises the HC signal mentioned in SLOW-IN. The HC signal is used as feedback to indicate to the SLOW-OUT circuits whether the typewriter is ready for a new code. All keys which are subject to solenoid activation except SPACE and PERIOD yield LEVEL signals from the TYPEWRITER OUTPUT MATRIX. SPACE and PERIOD raise HC independently of the levels to provide the necessary feedback signal (see sec. F-3e).

F-7e' The switch contacts associated with each key remain closed for the duration of the operation of the typing mechanism, hence supply an ideal feed-back signal. In the cases of TAB and CARRIAGE RETURN, different methods of providing switch contact are provided to render contact for the entire duration of the operation. In the case of TAB, two switches in parallel are required such that one or the other or both will be in contact for the duration of the operation.

F-7f A SPACE KEY, as opposed to the SPACE BAR, is used to shift the carriage one space. This key activates a typing lever as any other character key but no character is typed; the carriage merely shifts. The reason for avoiding use of the SPACE BAR is that it is designed to cause multiple spacing if held down more than a short time.

F-7g The SLOW-OUT system is designed to operate the typewriter at an average rate of approximately 8 characters/second (i.e., 1 character/4 drum cycles). The OB's and the "execute" pulse energize the associated TYPEWRITER relays by means of puller tubes located in the LOGIC CHASSIS (dwg. 64).

F-7h THE PUNCH: The PAPER TAPE PUNCH is a standard FLEXOWRITER punch unit. Reference is directed to FLEXOWRITER literature for a detailed description. In brief, the unit perforates tapes conditionally in 5 levels and unconditionally in the sprocket level whenever it is pulsed. The punching in 5 levels is a direct function of the contents of the OB flip-flops (which contain the output code). Whenever a frame is punched, the tape is advanced 1/10 inch by a sprocket wheel.

F-7i The punching is under the control of six 2D21 thyratrons located behind the PUNCH unit. Ref. Drawing 47: Five of these thyratrons control the LEVEL SOLENOIDS; the sixth (V6) controls the CLUTCH SOLENOID. Whenever the GRID #1 common becomes high (1 millisecond approx.), the CLUTCH SOLENOID thyratron is unconditionally fired and the other thyratrons fire as a function of the OB's.

F-7j When the CLUTCH SOLENOID is energized a mechanical cycle proceeds, punching the appropriate levels and sprocket and also advancing the tape. During this cycle a cam-controlled switch opens, removing B+ from the thyratron plate circuits extinguishing them at the appropriate time. An INTERLOCK switch can also remove B+ to prevent thyratron firing in the event that the tape supply reel is empty.

F-7k The CLUTCH SOLENOID may be energized continuously by means of a manually-controlled switch on the PHOTO-TAPE READER front panel. This causes the PUNCH to punch blank tape in an un-interrupted manner.

F-7l PUNCH PULSES (1 millisecond) are delivered to the PUNCH thyratrons by the OUTPUT SYSTEM at a rate of one every two drum cycles (i.e., 17/sec.) provided the OC configuration or the PUNCH switch (on the TYPEWRITER) calls for them. The PUNCH requires no synchronization. It will respond properly to each PUNCH PULSE provided the pulses do not exceed the maximum permissible rate. 17/sec is safely below that rate.

F-7m A detailed schematic of the punch driver circuit may be found on Drawing 51. SW1, both sections of which are shown on this drawing, permits the operator to turn the PUNCH MOTOR and B+ supply off. This switch may be turned off to avoid running the PUNCH MOTOR when programs being run do not call for any PUNCH OUT operations. The switch is located on the PUNCH CHASSIS itself.

F-7n CARD EQUIPMENT: The CA-1 CARD EQUIPMENT consists of an IBM 026 unit with adapting circuits. The adapting circuits are fed by the contents of the OB's by means of relay puller tubes located in the LOGIC CHASSIS (dwg. 64). Control signals also arise in the OUTPUT system as a function of the OC's and also energize relays by means of puller tubes. Since the CARD EQUIPMENT is optional accessory equipment and is a study in itself, no more details of the equipment itself will be given in this particular manual. Study of the G-15D circuits will reveal that characters are delivered to the CARD PUNCH at the rate of 1 character/3 drum cycles ---- (i.e., approx. 12 char./sec.).

SLOW-OUT LOGIC

F-8a In order to permit maximum flexibility in controlling the sequence and nature of codes to be supplied to the OB's, 8 different operations may be called for by means of the individual FORMAT CHARACTERS (instructions) in the FORMAT (sequence of instructions). Since both LINE 19 and AR can supply the output data, independent FORMATS may be established for each. The formats are stored in words 0-3 of LINE 2 (for LINE 19) and LINE 3 (for AR); the OC configuration will decide which FORMAT will be followed.

F-8b The 8 FORMAT CHARACTERS (instructions) are listed on Drawing 50. Below is an outline of activity resulting from each character. Assume that at the beginning of each code-to-OUTPUT cycle a FORMAT character will reside in a 3-bit static register, consisting of the OF flip-flops, and will be decoded and interrogated.

ACTION RESULTING FROM FORMAT CHARACTERS (Ref. Dwg. 52)

F-8c [DIGIT]$_{OF}$: The output line (M19 or AR) will be precessed through the four OA flip-flops for a recirculation cycle. This results in: (1) the highest order 4 bits (hex. character) residing in the OA's, (2) the original contents of the line shifting to the left 4 bit positions, and (3) all ZEROS in the lowest order 4 bit positions of the line. (The latter is accomplished by resetting the OA's prior to the precession.) After the highest order 4 bits are obtained in the OA's, Ⓜ transfers them to the lowest order four OB's, (OB1-OB4), simultaneously resetting the OA's. OB5 will be set to ONE at this time (if it is not already set) provided the OA data was non-ZERO or if the OC configuration or PUNCH switch called for a PUNCH TAPE or PUNCH CARD.

F-8d These conditions pertaining to OB5 control facilitate suppression of insignificant ZEROS. That is, if an all-ZERO code from the OA's is placed in OB1-OB4, and OB5 is not set, the resulting code (00000) will cause the typewriter to SPACE rather than TYPE "O". Whenever the PUNCH or CA-1 CARD EQUIPMENT is involved, the zero suppression circuit is disabled (by OC2+⟨MAN PUNCH⟩), and OB5 will be high for any hex. character. A ZERO will appear as 10000 in the OB's, hence "0" will be typed out whether significant or not. More will be mentioned on the subject of zero suppression as the other FORMAT CHARACTERS are discussed.

F-8e [SIGN]$_{OF}$: During SLOW-OUT operations the OS flip-flop monitors the sign bit position of the word being sent to output. In the case where AR constitutes the output line, the T1 position of AR is periodically monitored; when LINE 19 is the output line, the T1 position of word 107 of LINE 19 is monitored. This is under the control of the OC's and appropriate timing signals.

F-8f When the FORMAT CHARACTER calls for SIGN, if the OS flip-flop has been set, OB1 will be set at Ⓜ time; the remaining OB's will all remain reset. This will yield a 00001 configuration which is the MINUS SIGN code. Had the OS flip-flop been in the reset state at this time (representing +), the OB's would have assumed a 00000 configuration, representing a SPACE.

F-8g No AR or LINE 19 precession takes place during this operation. When a SIGN-to-OUTPUT is called for, the word involved should be in its initial state (i.e., not shifted) to assure that the sign position actually contains sign information rather than some other bit which appeared there as the result of previous shifting operations. The OS flip-flop merely monitors whatever bit happens to be in the sign position whether or not it actually represents a sign.

F-8h [CR+TAB]$_{OF}$: When either a TAB or CARRIAGE RETURN constitutes the FORMAT CHARACTER, the same SLOW-OUT activity takes place, the only difference being in the exact code sent to the OB's for control of the output medium. A CR or TAB is ordinarily called for at the end of every word transmitted to the OB's. At this time, normally a total of 28 bit positions of shift have been experienced by the line (M19 or AR) supplying the output information. An additional bit of shift is appropriate for positioning purposes only.

F-8i The operation calls for a 1-bit precession of the appropriate output line through OA1 only, and lasting for one recirculation time of the line in question (M19 or AR); also, at Ⓕ time, the FORMAT CHARACTER itself is copied from the OF's to OB1-OB3. OB4 and OB5 will be reset. As a result, a 00011 configuration will appear in the OB's as a result of a TAB or 00010 in the case of CARRIAGE RETURN.

F-8j [STOP]$_{OF}$: The last character of a FORMAT should be STOP (sometimes called END). In the case of AR feeding OUTPUT, the STOP code in the OF's is copied into the OB's yielding the 00100 configuration in the OB's -- the STOP code. Also the OC flip-flops will be reset, terminating the OUTPUT operation.

F-8k If LINE 19 feeds the output, as soon as the STOP code appears in the OF's, LINE 19 is interrogated. If during the course of one drum cycle any ONES are detected in LINE 19, OF1 is set, converting the STOP code to a RELOAD code. As LINE 19 is precessed and its contents sent to output, ZEROS are inserted in its lowest order end. This is accomplished every time a precession takes place by resetting the OA's prior to each precession. When the contents have been "emptied into output" and there is no more information left, this is an appropriate time to terminate the OUTPUT operation. If LINE 19 is "empty", the STOP code will remain in the OF's, be copied into the OB's, and the OC's will be reset (as in the case of AR to OUTPUT).

F-81 In the event that upon interrogation of LINE 19, a "1" converted the STOP code to a RELOAD code, the procedure will follow that outlined below.

F-8m $[\text{RELOAD}]_{\text{OF}}$: When a RELOAD code appears in the OF's, at the time of interrogation (i.e., after the conversion process above), the code itself is sent to the OB's yielding a 00101 configuration. Also the OD flip-flop is set, which causes a repetition of the entire FORMAT (i.e., after the RELOAD operation is performed, the FORMAT CHARACTERS will be interrogated again, starting with the first character.

F-8n RELOAD codes in the OF's normally arise only as the result of a converted STOP code. RELOADS placed in a FORMAT by the programmer would result in an OUTPUT operation which could not terminate itself. RELOADS can be used in conjunction with a "SET READY" command for certain special purposes, but a thorough understanding of the conditions and timing involved in SLOW-OUT is required if this type of programming is contemplated.

F-8o $[\text{PERIOD}]_{\text{OF}}$: This calls for copying the OF's into OB1-OB3, yielding the 00110 configuration in the OB's. This is for the purpose of editing typed copy. As a byproduct of the operation, OB5 is set when OB1-OB4 are cleared (after the response to the PERIOD code by the output medium). This stops zero suppression so that any "O" code following the period will be typed as "O".

F-8p $[\text{WAIT}]_{\text{OF}}$: This is strictly an editing function to cause the typewriter to ignore (skip) a hex. character in the output line. WAIT calls for a 4-bit precession just as a [DIGIT]OF does; however, the character obtained in the OA's is ignored. Instead of sending the character in the OA's to the OB's, the WAIT code in the OF's is copied into the OB's, yielding the 00111 configuration. The TYPEWRITER will not respond, but the PUNCH will punch 00111. (Should the tape be used for INPUT, the WAIT code will compensate in INPUT for the 4-bit precession in OUTPUT by inserting a hex. ZERO in LINE 23.)

F-8q SUMMARY OF FORMAT CHARACTER ACTION: A brief outline of the activity mentioned above is provided on Drawing 53. Drawing 46 illustrates the response of the PUNCH and TYPEWRITER to the codes in the OB's assuming that an "execute" pulse is delivered to the medium. A space punched on PAPER TAPE is insignificant since it represents only a frame of blank tape.

F-8r Standard FORMATS are illustrated on Drawing 50. The FORMAT used for LINE 19 output yields a tape, the sequence of characters upon which renders it compatible with the INPUT SYSTEM. Any FORMAT sequence may be written by the programmer and is limited only by his imagination and/or computer restrictions.

F-8s (note that FORMAT CHARACTERS listed on dwg. 50 appear in reverse as to order of digits when appearing in the OF's or the lowest order three OB's (dwg. 53). This is a function of the circuit wiring and is a minor technicality resulting in no logical discontinuity--the placement of bits is only arbitrary anyway.)

SLOW-OUT TIMING

F-8t (Ref. dwg. 52 - bottom) Primary timing signals are provided by flip-flops OZ, OY,·OE, and OG. Configurations of these flip-flops qualify gates which provide other timing signals. Activity of these circuits is illustrated on the timing diagrams shown on Drawings 54 and 55.

F-8u A basic character output cycle involves two drum cycles beginning with the $OY\cdot\overline{OE}$ configuration and concluding with the $OY\cdot OE$ configuration. In the event that the TYPEWRITER or CARD PUNCH is involved in the OUPPUT operation, time is "killed" between character cycles in the $\overline{OY}\cdot\overline{OE}$ configuration to the extent of an integral number of drum cycles as a function of OC and feedback control as follows:

PUNCH: no delay; OY is always high.

CARD PUNCH: one drum cycle delay. This is introduced by "TYPE" signal (present if OC's = 1011) which causes OY to be reset at the conclusion of each basic cycle yielding $\overline{OY}\cdot\overline{OE}$. One drum cycle later OY is set ($\overline{HC}$ is high) establishing $OY\cdot\overline{OE}$ which begins a new cycle.

TYPEWRITER: two or more drum cycles delay. "TYPE" (present with OC's = 1000 or 1001) causes OY to be reset, yielding $\overline{OY}\cdot\overline{OE}$. When the TYPEWRITER is executing a mechanical cycle, HC will be high until it is concluded. Until HC drops and $\overline{HC}$ becomes high, OY cannot be set. If DIGIT, SIGN, CR, or TAB is the FORMAT CHARACTER being processed, $\overline{OF3}$ (function of the FORMAT CHARACTER) will prevent OY from being set after one drum cycle delay (regardless of TYPEWRITER feedback) by holding HC high. After one drum cycle delay, $TYPE\cdot\overline{OY}\cdot\overline{OE}\cdot TO$ sets OF3, dropping HC, provided TYPEWRITER switches do not hold it high. Once HC drops, OY can be set, starting a new cycle. HC may be kept high indefinitely by <SA>, hence the ENABLE switch provides a means of interrupting a TYPE OUT or CARD PUNCH.

F-8v At the beginning of each cycle ($OY\cdot\overline{OE}$), OG controls the circuit which obtains the new FORMAT CHARACTER in the OF's. The character is interrogated and its process is executed. The OB's do not receive the output code until the conclusion of the 2 drum cycle period, and the "execute" pulse is not delivered to the medium to cause response until the beginning or the following character cycle. Therefore while the output medium is responding to one code, the SLOW-OUT circuits are obtaining the next.

OBTAINING FORMAT CHARACTERS

F-8w The FORMATS are stored in words 0-3 of LINES 2 and 3. The OC's determine the line from which FORMAT should be taken. When the SLOW-OUT operation is initiated, OD is set. The first OG output will cause the appropriate line (M2 or M3) to be read via the three OF flip-flops into the four-word line, MZ, during word times 0-3. (OD, in conjuction with the OC's, sets up the path; OG determines the time.) As a result of this, the highest order 3 bits of word 3 (of LINE 2 or 3) will reside in the OF's and the remaining information (originally in words 0-3 of LINE 2 or 3) will reside in MZ, displaced 3 bits to the left. The lowest order end of MZ, containing information to the right of the STOP code, is logically unimportant. The operation does not disturb the contents of the selected format line (M2 or M3).

F-8x This first FORMAT CHARACTER in the OF's will be interrogated and the process for which it calls will be executed. When the next FORMAT CHARACTER is called for (when OG rises), OD will be in the reset state (unless the first format character was a RELOAD--which is ridiculous). OG·$\overline{\text{OD}}$ will cause MZ to precess through the OF's obtaining the next 3-bit character in the OF's and shifting MZ to the left 3 more bit positions.

F-8y This activity repeats until a RELOAD code appears in the OF's. A RELOAD sets OD causing the initial process to be repeated.

F-8z (Note that OG is set by a T1 pulse of word time 0 and reset by TF·OG. This means OG will be high during word times 0-3 except for the T1 period of word time 0. This is one pulse period short of the desired four word times; however, this will only affect MZ to the extent that the T1 position of word 0 will always contain whatever was originally in that bit location of MZ. It will not affect the FORMAT CHARACTERS and their positioning in any respect.)

F-8aa At this point the reader should be able to trace through any SLOW-OUT activity by referring to Drawings 52 - 55. In Drawings 54 and 55 the key signals are shown in reference to the time axis - an important factor in understanding the activity. The reader is advised to draw colored lines between corresponding TO dots at the top and bottom of each drawing. The signals are labelled at the left in abbreviated form. (For instance, the signal Ⓜ = Ⓝ·$\overline{\text{OE}}$·SLOW-OUT, where Ⓝ = T1·OZ. The label on dwg. 55 defines Ⓜ = $\overline{\text{OE}}$·OZ·T1, which is incomplete. SLOW-OUT also qualifies Ⓜ; however, the reader is expected to take such a term for granted since the operation is a SLOW-OUT. When in doubt, consult the more detailed drawings.)

MISCELLANEOUS (SLOW-OUT):

F-8ab (1) ⟨A⟩⟨SA⟩·Ⓝ sets OD in addition to setting the OC's. This is the manual method of initiating TYPE OUT AR. OD must be set by the above signal since S2·DS will not be present to do so as in the case of program-initiated SLOW-OUT operations. Also, until the ENABLE switch is turned off, no type-out will occur since HC will be held high by ⟨SA⟩.

F-8ac (2) When the PUNCH is the output medium, one frame of blank tape will be punched prior to the first data element in the OB's. Suppression of this initial insignificant PUNCH "execute" pulse would have involved extra cost to conserve absurdly small amounts of paper tape. OD inhibits the corresponding TYPEWRITER "execute" pulse to prevent a SPACE, but in so doing inhibits the "execute" pulse when RELOAD is the code in the OB's. This is tolerable in the case of the TYPEWRITER since it does not respond to RELOAD anyway; in the PUNCH case it would be intolerable since RELOADS must be punched to render the output tape compatible with the INPUT SYSTEM.

F-8ad (3) After a STOP code has been obeyed as the final FORMAT CHARACTER, the three bits that follow it in the format line will appear in the OF's. $[\overline{\text{STOP}}]_{OB}$ is applied to the Ⓑ, Ⓘ, and Ⓟ terms to prevent simultaneous set and reset terms from being applied to the OB's at OC_r time. This could result in malfunction of a subsequent PUNCH operation. The most fool-proof method of writing formats is to follow the STOP code with a SIGN code.

F-8ae (4) In any of the cases in which the OC configuration calls for a TYPE-OUT, an "execute" pulse may also be delivered to the PUNCH if the PUNCH SWITCH on the typewriter is in the ON position - raising <MAN PUNCH> to OV. In such a case, the output speed is a function of the typewriter and the ZERO-suppression is disabled.

FAST-OUT - GENERAL

F-9a The active FAST OUT operations only involve optional auxiliary equipment--MAGNETIC TAPE and devices including high-speed punches.

F-9b READY: The READY STATE is not an operation but is classified as a FAST OUT "operation" by virtue of its OC configuration (0000). READY is high when all the OC's and OD are reset. The signal "READY" lights a front panel neon "R" in an unusual way. Most neon exciting signals in the G-15D involve approximately a 60V difference between ON and OFF. The READY signal involves a difference of 20V since it is the output of a conventional AND gate. This signal (OV or -20V) is used to turn a neon exciter triode on or off. This triode is located in the LOGIC CHASSIS (dwg. 64).

MAGNETIC TAPE WRITE (FAST-OUT)

F-10a This FAST-OUT operation delivers data to the magnetic tape unit(s) at the peak rate of 463 characters/second. There is no flexibility in output format as there is in the SLOW-OUT system. The sequence in which data is sent from LINE 19 to the MAGNETIC TAPE equipment is:

> 29 HEX. CHARACTERS followed by either a RELOAD or a STOP depending upon whether or not LINE 19 is empty.

F-10b The flexibility in SLOW-OUT was available since the individual output code cycles were of long duration (minimum of 2 drum cycles). However, there is no need for flexibility in the FAST-OUT sequence since the fixed format which is followed supplies a code sequence to MAGNETIC TAPE which is compatible with the INPUT SYSTEM. The purpose of the MAGNETIC TAPE equipment is primarily storage. That is, if a 108 WORD BLOCK is written on the tape from LINE 19, it may be restored to LINE 19 later by an INPUT operation. The fixed format fulfills this requirement.

F-10c In general, the sequence calls for "unloading" LINE 19, in 4-word groups, onto MAGNETIC TAPE. The manipulation of the 4-word groups is almost a SLOW-IN operation in reverse.

F-10d Ref. dwg. 61: To facilitate the sequence above, LINE 19 is precessed through MZ for 108 word times (starting with word time 0). This results in: (1) the original contents of words 104-107 of LINE 19 residing in MZ, (2) the original contents of LINE 19 being shifted left 4 words, and (3) the lowest order 4 words of LINE 19 being filled with ZEROS (original contents of MZ - rendered all ZEROS by proper programming of the initiating command).

F-10e Once MZ has extracted the highest order 4 words from LINE 19, MZ is copied into LINE 23, freeing MZ to obtain the next highest order 4 words of LINE 19 (synchronized on the next TO pulse).

F-10f With the original highest order 4 words now in LINE 23, precessions of LINE 23 through the OA's proceed on alternate 4-word recirculation cycles. During the 4-word recirculation cycles between precessions, the OA's will contain 4-bit configurations starting with the highest order hex. character and progressing sequentially to the lowest order hex. character. During the time that these hex. characters are static in the OA's, an "execute" pulse permits AND gates to send them to the writing circuits in the MAGNETIC TAPE equipment (see dwg. 60 - top). These "execute" pulses last 4 word times or approximately one millisecond. This takes care of LEVELS 1-4. LEVEL 5 is written any time the information written on tape is a hex. character.

F-10g A means is provided to determine when 29 hex. characters (i.e., the entire contents of LINE 23) have been written on tape. This involves turning OA2 on (by Ⓩ) prior to the precession which obtains the first hex. character in a 29 character group. This is known as the MARKER BIT. As a result of this, after the first precession, the 0100 configuration will have been written in the lowest order 4 bit positions of LINE 23. Prior to any other precession the OA's are reset to 0000.

F-10h During the course of the first 29 precessions, the MARKER BIT established in OA2 (by Ⓩ) will be transmitted to the writing station of M23. Only after the 29th precession (i.e., during the 30th) will the MARKER BIT fail to reach M23w. OF1 monitors OA4 during each precession. During the first 29 precessions OF1 is set, and remains set between precessions causing 5th LEVEL to be written. After the 30th precession, OF1 remains down, failing to write 5th LEVEL. The configuration in the OA's at this time is 0100. Since 5th LEVEL is not written, the tentative configuration of the LEVELS to be written on tape is 00100; however, $OB3\cdot\overline{OF1}$ can qualify the LEVEL 1 gate which could result in a 00101 configuration.

F-10i For a period of more than one drum cycle prior to establishment of the above code, an $OB3_s$ gate searched LINE 19 for ONES. If LINE 19 contained any ONES, OB3 would be set; if it contained no ONES, OB3 would remain reset. Since OB3 controls writing of LEVEL 1 at this time, and LINE 19 controls OB3, an "empty" LINE 19 will cause a 00100 configuration (STOP code) to be written; otherwise 00101 (RELOAD code) will be written. This resembles the activity in the SLOW-OUT system when $[\mathrm{STOP}]_{OF}$ is the FORMAT CHARACTER; the means, however, differ considerably.

F-10j Had the RELOAD code been written, the next 4-word group residing in MZ (resulting from the second M19 - MZ precession) would be copied into LINE 23 and the 29 HEX. CHARACTER - STOP/RELOAD cycle would repeat.

F-10k These 4-word cycles will repeat until LINE 19 is exhausted of information. Normally 108 words of LINE 19 will contain data, hence 27 4-word cycles will occur. If the tape-recorded data from LINE 19 is, at some future time, to be read back into LINE 19 in the same relative word positions, at least one ONE should exist in the original lowest order 4-word group of LINE 19. Simple programming can assure this. The operation is concluded after the STOP code is written and the OC's are reset, establishing READY.

F-10l Drawing 60 is the FAST-OUT block diagram covering the MAGNETIC TAPE WRITE case. Drawing 61 provides an example of a writing operation in which only the highest order 8 words of LINE 19 contain data. Beyond this, little can be said about the data handling logic.

F-10m MAGNETIC TAPE motion control as applied to MAGNETIC TAPE WRITE is a simple matter as far as the G-15D circuits are concerned. Ref. Drawing 60 - top: DS·SO·SV (high during the time of TRANSFER associated with the initiating command) will fire the FORWARD thyratron in the electro-mechanical tape transport system. The tape will be accelerated to its normal speed of 7.5 in/sec in approximately 5 milliseconds. How soon data will appear to be written on the tape is a function of the word time during which the initiating command is in the TRANSFER state; this should be WORD 0 to assure maximum leader.

F-10n The tape transport mechanism is stopped by the READY signal which fires the STOP thyratron.

F-10o The MAGNETIC TAPE recording mode is similar to that employed by the drum in that only ONES are written and ZEROS are erased magnetic surface. Any section of tape which is selected for recording must be erased. This is accomplished by bulk erasure.

MAGNETIC TAPE READ CONTROL (SLOW-IN)

F-11 MAGNETIC TAPE has already been mentioned as a source of INPUT data; however, its control was not mentioned at that time. Ref. Drawing 47 - lower right: Control of the mechanism is similar to the MAGNETIC TAPE WRITE case. DS·S3·SV (high during the TRANSFER state of the initiating command) fires the FORWARD thyratron. The tape will move past the READ-WRITE head resulting in codes being read into the INPUT system. Finally, when a STOP code reaches the OB's, Ⓢ · $[\mathrm{STOP}]_{OB}$·OC1 fires the STOP thyratron; this minimizes tape travel after the STOP code is detected. (The READY signal, which normally stops the tape, will not appear until 1 - 2 drum cycles after the STOP code is received since this much time must be allowed for the MZ - M19 precession associated with a STOP code.) There are no restrictions on TRANSFER timing. Once the tape is in motion (7.5 in/sec.), data previously recorded on it will appear in terms of LEVEL signals at the same character rate that was established in the writing operation. (See also sec. F-15d.)

MULTIPLE MAGNETIC TAPE UNITS

F-12a CHARACTERISTIC qualification: Up to four MAGNETIC UNITS may be attached to one G-15D. The one selected to be active during a MAGNETIC INPUT-OUTPUT operation is a function of the CHARACTERISTIC bits of the initiating command. The CHARACTERISTIC to which a unit will respond is selected by a 4-position switch on the unit itself as shown on Drawing 47.

F-12b When more than one MAGNETIC UNIT is attached to a G-15D, the units are "chain-wired". Input and output wires are connected to two connectors on each MAGNETIC TAPE UNIT. In this way, the G-15D itself only requires one connector for MAGNETIC equipment.

F-12c The LEVEL output stages of each unit are cathode-followers. Connecting the outputs of each unit in parallel constructs cathode-follower OR gates, hence no additional mixing circuits are required. Only one unit should be operated as a reader at any one time; however, more than one can be simultaneously writing in the event that duplication is desired.

MAGNETIC TAPE WRITE FILE CODE

F-13 The 6th LEVEL on MAGNETIC TAPE is available for FILE CODES, which may be used to locate information blocks or groups of blocks. Just where these FILE CODES are written is a function of the controlling program in the G-15D. Once written, a FILE CODE may be used as the object of a SEARCH operation (for tape positioning purposes). Writing a FILE CODE does not involve tape motion. DS·S7·SW, which arises during the TRANSFER state of the WRITE FILE CODE command, qualifies the 6th LEVEL writing circuit in the magnetic unit selected by the CHARACTERISTIC code. TRANSFER must last for 4 word times to yield a 1 millisecond writing pulse; there are no restrictions as to when these word times can occur during a drum cycle.

MAGNETIC TAPE SEARCH

F-14a TAPE SPEED: 0100 or 0101 in the OC's yields the signal "FAST". This turns on a pair of relay puller triodes in the tape unit energizing the FAST relay (see dwg. 47). This relay reconnects the windings of the multi-pole capstan drive motor in the tape transport mechanism such that its angular velocity is increased by a factor of 6:1. This means that the tape, when in motion, will be driven at a rate of 45 inches/sec (i.e., 6 x 7.5 in/sec).

F-14b FORWARD or REVERSE: Regardless of direction, when the tape reaches a FILE CODE, a 6th LEVEL output will arise. If the TRANSFER state of the initiating command is not still high, the 6th LEVEL output signal will set OF3 which will reset the OC's. This yields READY, which fires the STOP thyratron.

F-14c The $\overline{DS \cdot S1}$ term, which qualifies the $OF3_s$ gate, permits the system to ignore FILE CODES existing close to the point on the tape from which the search starts. By programming the duration of TRANSFER the programmer can render the search selective to different degrees.

F-14d FORWARD SEARCH: DS·S1·SV (if qualified by CHARACTERISTIC) fires the FORWARD Thyratron; READY fires the STOP thyratron.

F-14e REVERSE SEARCH: DS·S1·SU (if qualified by CHARACTERISTIC) fires the REVERSE thyratron; READY fires the STOP thyratron.

TIME ALLOWANCES - MAGNETIC TAPE

F-15a In general, no change in motion of the tape should be called for within less than a drum cycle of the previous change. The reason for this is that the charges on the condensers linking the thyratron plates in the MTA-2 need time to stabilize.

F-15b A MAGNETIC TAPE READ or WRITE operation should not follow a SEARCH operation by less than 16 drum cycles, otherwise the capstan will not have decelerated to 7.5 inches per second peripheral speed. This holds true even if different MAGNETIC TAPE UNITS are involved since all capstan motors are subject to FAST control regardless of CHARACTERISTIC code.

F-15c A MAGNETIC TAPE READ should not follow a MAGNETIC TAPE WRITE by less than 4 drum cycles. A writing operation saturates the reading amplifiers in the MAGNETIC TAPE UNIT and recovery time must be allowed. As a matter of fact, no INPUT operation should immediately follow a MAGNETIC TAPE WRITE since the MAGNETIC TAPE reading circuits, until stabilized, will disturb the INPUT LEVELS and HC.

F-15d An MTA-2 can be programmed to search in reverse for a STOP code by first calling for a REVERSE SEARCH (04 → 31) followed shortly by a GATE TYPE-IN (12 → 31). The tape will travel in reverse at 7.5 inches per second until a STOP code is reached. The STOP code will cause READY to rise in 1-2 drum cycles; READY fires the stop thyratron. $\textcircled{S}\cdot[\text{STOP}]_{OB}\cdot OC1$ will not fire the stop thyratron (as per Sec. F-11) since OC1 is low under these circumstances. The purpose of OC1 in the above term is to allow ample "overshoot" so that the STOP code may be reliably read in a forward direction if required. (Programming literature should be consulted for rules relating to this special use of the MTA-2.)

FAST PUNCH OPERATIONS (FAST-OUT)

F-16 Two FAST-OUT operations (S = 02, 03) are available for controlling and feeding data to optional peripheral devices such as the PTP-1, AN-1, and AN-2. These operations use the FAST-OUT logic, controlled by feedback, to supply characters from LINE 19 to the output devices at rates determined by the devices themselves. (Details are discussed in literature dealing with the accessories in question.)

"SET READY" COMMAND (00 → 31)

F-17 The SET READY command resets the OC flip-flops (ref. dwg. 45: DS·SO·SU gate on OC_r). It also sets OD if the OC configuration at the time of execution is not divisible by 4 (ref. dwg. 60, lower right: DS·SO·(OC1 + OC2) on OD_s). The combination of OD·FAST-OUT results in a 4-word precession of LINE 19, followed by READY, which rises after the second TO following execution of the SET READY command. This feature is quite useful in programming. SET READY is also used in command sequences for purposes of positioning MAGNETIC TAPE.

SECTION G

POWER SUPPLIES AND TURN-ON CYCLE

The circuits described in this section introduce no new logical concepts, but do require a reasonable knowledge of some circuits previously described.

POWER SUPPLIES

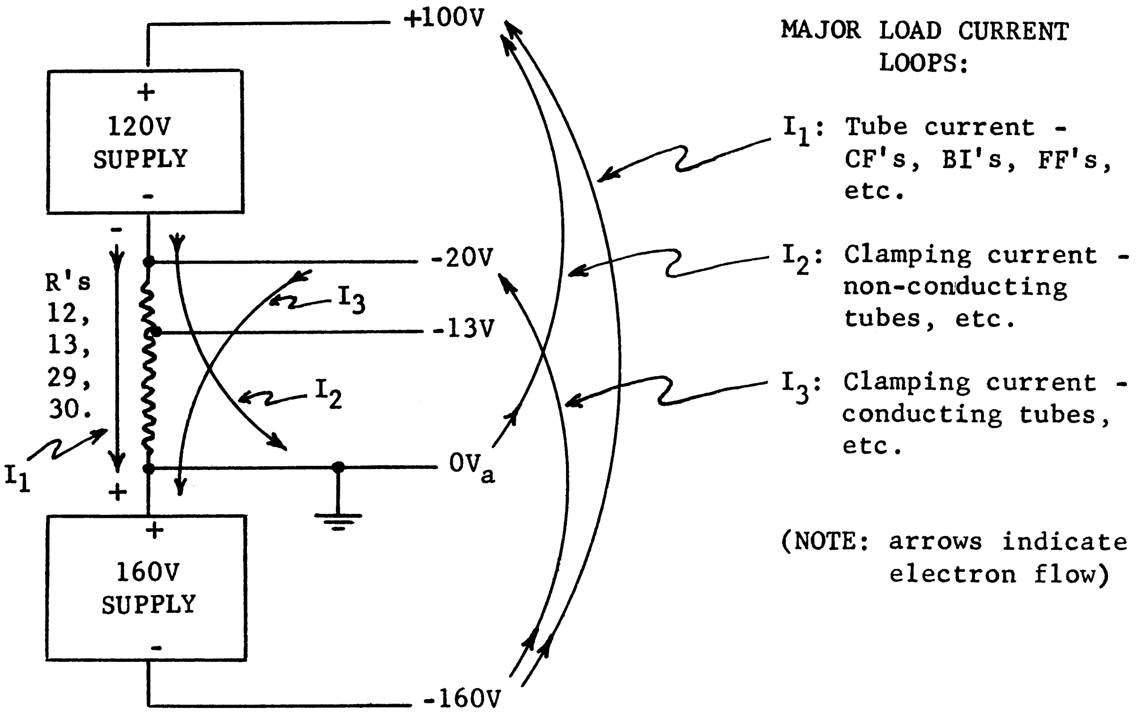
G-1a Four D.C. power supplies supply all of the necessary D. C. potentials to operate the computer. These supplies will yield their outputs, provided their primary circuits are connected to A.C., by a relay (K3) associated with the TURN-ON system. The four supplies are all conventional full-wave bridge rectifiers terminated with choke input filters and yield outputs as follows:

+ 250V for writing pentodes and the CLOCK CHASSIS

+ 160V for relays and solenoids

+ 120V }
- 160V } for +100V, 0V, -13V, -20V, and -160V to supply the logic circuits (The -25V and -75V supplies for READ CLOCK are parasites on the +100V and -160V supplies - ref. B-3h-j.)

G-1b The +120V and -160V supplies are interconnected in terms of load current loops and form the circuit shown below:



MAJOR LOAD CURRENT LOOPS:

I_1: Tube current - CF's, BI's, FF's, etc.

I_2: Clamping current - non-conducting tubes, etc.

I_3: Clamping current - conducting tubes, etc.

(NOTE: arrows indicate electron flow)

G-1c This hookup yields four stable output voltages by means of only two supplies. Drawing 58 shows the details of the arrangement.

G-1d The ground returns for +250V, +160V, and [+120V and -160V] are $0V_c$, $0V_b$ and $0V_a$ respectively. They are all at 0V as far as D.C. is concerned but not necessarily in terms of RF (i.e. transients). By using separate ground returns, the transient problem is reduced.

G-1e The average load presented by the logic circuits is steady since, on the average, whenever one tube cuts off, another turns on. Deviations from this average may be expected to be of very short duration. At any rate, generous amounts of filter capacitance shunt every supply, rendering them tolerant of changing load requirements and relatively immune to line voltage transients unless they are of unreasonable proportions.

G-1f Since these supplies are unregulated, their output levels are a function of line voltage. Line voltage is adjustable by means of a variable transformer (front panel control), the output of which supplies a boost to the line by means of T4 (dwg. 56 - left). The adjusted (boosted) AC feeds the primary circuits of the power supply transformers and the filament transformer. The motors employ AC directly from the line since they are not critical. The variable transformer is adjusted for a 6.3V indication on the filament voltmeter.

G-1g Gradual drifts in line voltage may be compensated for by manually adjusting the variable transformer; however, should this become a problem, a line voltage regulator is recommended. Should the line voltage be subject to unreasonable dips and/or interruptions, rotary machinary (M-G set) is recommended. The G-15D is tolerant of sizable deviations in line voltage, but it has its limits. The limits are a function of how effective the scheduled maintenance activities are. Ordinarily a G-15D should be expected to tolerate at least 10% line voltage deviations.

G-1h To facilitate scheduled maintenance (marginal checking) activities all supplies are variable by means of front panel controls. Varying the supplies individually can upset circuit constants to the extent that a deteriorating component may be detected and replaced before it fails in normal service. Varying the line voltage transformer reduces all supplies in proportion and also the filament voltage, providing another means of marginal checking.

G-1i A per-cent reading meter is available to monitor all potentials. A switch is provided to connect the supply to be measured in the correct polarity and with the proper amount of multiplier resistance.

G-1j Varying of supplies is accomplished as follows:

+250V:	Control C (variac), continuously variable.
+160V:	Control B (switch - SW3), 10% drop.
+100V:	Control A (switch - SW2), 10% drop.
-13V:	Control D (potentiometer - R12), continuously variable.
-20V:	Control E (potentiometer - R13), continuously variable - affects - 13V in proportion
-160V:	Control F (switch - SW1), 10% drop.

G-1k Controls A and F, when inserting resistance in series with the loads of their associated supplies (to create 10% drop), also adjust the load on the supply by compensatory resistance to keep power supply current constant. This is necessary since the currents of the +120V and -160V supplies determine the level of the -20V and -13V potentials.

THE TURN-ON CYCLE

POWER

G-2a Drawing 56 illustrates the TURN-ON CYCLE activity and the hardware involved. When power is first applied to the computer, TIMING-MOTOR #1 (TM1) runs, driving a built-in gear train. The output shaft drives a rotary 2-wafer switch which controls relays K1 and K2 as shown on Drawing 56 (upper right). TM1 stops when the K1·K2 configuration is achieved. K1 and K2 control the amount of series resistance in the filament transformer primary circuit, incrementing the filament voltage as shown on Drawing 56 (upper left).

G-2b The system is relaxed when TM1 stops, but is ready to proceed if the D.C. RESET button is pushed. D.C. RESET starts the second phase of the TURN-ON-CYCLE by starting TM2. TM2 with its gear train also drives a rotary switch. This switch has 4 wafers. Shortly after TM2 starts, K3 pulls in, energizing all D.C. power supplies. When this happens, the D.C. RESET button can be released.

G-2c Safety circuits can open K3 and turn off D.C. K5, K6, K7, and K8 provide overload safety features. Also pairs of jumpered points located in each DIODE CLAMP package (points L and M) are chain-wired such that removal of any DIODE CLAMP package will break the chain, opening K3 and turning off D.C. (Naturally, D.C. should have been turned off - by the D.C. OFF button - prior to removal of such a package).

TURN-ON LOGIC (Ref. dwgs. 56, 57)

G-2d As soon as D. C. appears, the ⟨CLEAR⟩ signal rises to make sure the OC's are reset immediately in order to prevent random activation of an INPUT-OUTPUT device. ⟨CLEAR⟩ also resets CQ to assure the first command to be read will be read at word N, rather than N+1 (dwg. 30). (See also sec. C-19e.)

G-2e Then $\langle\overline{\text{CLEAR}}\rangle$ drops to -20V, blocking recirculation of the NUMBER TRACK (CN) to clear it of any ONES.

G-2f Then the signal $\langle\overline{\text{O}}\text{P}\rangle$ drops to -20V and ⟨OP⟩ becomes high, resetting the CY flip-flop in the ORIGIN PULSE (i.e. TO) generating circuit to be found on Drawing 57. CY resets at the first clock pulse to detect ⟨OP⟩, and after many drum cycles it will be set when the first clock pulse detects $\langle\overline{\text{O}}\text{P}\rangle$.

G-2g While CY was set it caused CE to be set. With CY reset, CE remains set since nothing can reset it - yet. This $\overline{\text{CY}}$·CE configuration permits gate A to write ONES in every T29 position of CN, provided a ZERO is read at the READING FLIP-FLOP ($\overline{\text{CN}}$). When the 109th T29 pulse prepares to write a ONE, the very first ONE which was written appears at the READING FLIP-FLOP (CN) dropping the $\overline{\text{CN}}$ signal and preventing the writing by gate A.

G-2h CN, now present, causes CE to be reset, qualifying recirculation gate B. The information now recirculating consists of ONES in all but one of the

T29 positions. That particular T29 position containing a ZERO may be defined as "TO" (= $T29 \cdot \overline{CN}$), the ORIGIN PULSE. Gate B will continue to provide recirculation until $\langle \overline{OP} \rangle$ sets CY; then gate C takes over and the activity of gate B is of no consequence.

G-2i As soon as CY was set, rendering gate C the recirculation path, CE assumed its normal function of being high during EVEN word times and low during ODD. Note that CE cannot be reset at $T29 \cdot \overline{CN}$ (i.e. TO) time therefore it will inevitably be high during WORD 0; this is the "locking-in" system assuring CE's proper orientation.

G-2j From this point on, the basic timing signals required to operate the SLOW-IN system are established (i.e. TO and TF).

G-2k ⟨OP⟩ also resets CG, assuring that the first command to be read will not come from AR (dwg. 22). (See also sec. D-16b.)

G-2l The next operation of the TURN-ON CYCLE is the generation of the ⟨AUTO TAPE START⟩ signal. When this signal arises, it sets the OC's to 1111, initiating a PHOTO TAPE READ operation, resulting in LINE 19 obtaining the first block of information from the tape. This should be the desired contents of the NUMBER TRACK. Also, CQ and CG are reset and CH is set.

G-2m The next signal to arise from the motor-driven switch is ⟨NT⟩, which qualifies gate D (dwg. 57). Gate D copies the contents of LINE 19 into the NUMBER TRACK. It is not necessary to block recirculation of the NUMBER TRACK during this operation since it was cleared by dropping $\langle \overline{CLEAR} \rangle$, and the T29 information just established in it is the same as the new information to be copied from LINE 19.

G-2n The next operation in the TURN-ON CYCLE is another automatic PHOTO TAPE READ. This loads the next block of tape information into LINE 19 (and LINE 23), simultaneously making sure the COMMAND LINE is set to 23 and N = 00 (see dwg. 22). This last block of information can be a "LOADING ROUTINE".

G-2o After this operation is concluded K4 pulls in, stopping TM2 and lighting the READY light (green).

G-2p An interlock is provided to stop TM2 if its associated rotary switch needs stopping (to prevent premature signals) while the PHOTO TAPE READER is still operating. The condition of the FORWARD relay in the PHOTO TAPE READER provides this feature.

G-2q From the time the READY lamp lights, the computer is ready for operation. (Do not confuse this READY light with the READY ("R") neon on the neon panel.)

MAINTENANCE SWITCHES

G-3a These switches were mentioned in sec. E-12. Half of them duplicate the functions which are ordinarily provided automatically by the switch sections driven by TM-2. The switches are as follows:

CLEAR NT: duplicates the function of $\langle \overline{CLEAR} \rangle$. That is, it blocks recirculation of the NUMBER TRACK, clearing it of any ONES.

It does so by shorting the true output of CN to -20V; unusual, but effective.

SET OP: duplicates the functions of ⟨OP⟩ and $\langle\overline{\text{OP}}\rangle$ in controlling CY, thereby establishing ORIGIN PULSE (TO) in the NUMBER TRACK.

SET NT: duplicates the function of ⟨NT⟩, applying the LINE 19 reading flip-flop (M19) to the writing station of the NUMBER TRACK.

CLEAR M23: when depressed, drops $\langle\overline{\text{M23 CLEAR}}\rangle$ to -20V, thereby blocking recirculation of LINE 23 - clearing it of any ONES. (dwg. 3D290)

CLEAR M19: when depressed, raises ⟨M19 CLEAR⟩ to OV, thereby blocking recirculation of LINE 19 - clearing it of any ONES (dwg. 3D290)

SET M19: when depressed, raises ⟨M19 SET⟩ to OV, thereby qualifying the M19 writing gate ⟨M19 SET⟩·CN, copying the NUMBER TRACK into LINE 19. This does not block M19 recirculation, therefore "CLEAR M19" should be depressed first (dwg. 3D290)

G-3b (The D.C. LOCKOUT toggle switch is a safety switch which renders application of D.C. voltage impossible - ref. dwg. 56. It is particularly useful in preventing complete machine "turn-on" while maintenance procedures are being undertaken.)

PART II - SIMPLIFIED DRAWINGS

This collection of simplified drawings constitutes Part II of this manual. Drawings 1-64 refer to both NUMERIC and ALPHANUMERIC G-15s, while drawings 65-88 refer to ALPHANUMERIC models only.

INDEX OF PRINTS

NUMERIC G-15

ALPHANUMERIC IN/OUT SYSTEM

SOURCE - DESTINATION TABLE

S CODE	SOURCE
00 (OU)	Line 0
01 (OV)	" 1
02 (OW)	" 2
03 (OX)	" 3
04 (1U)	" 4
05 (1V)	" 5
06 (1W)	" 6
07 (1X)	" 7
08 (2U)	" 8
09 (2V)	" 9
10 (2W)	" 10
11 (2X)	" 11
12 (3U)	" 12
13 (3V)	" 13
14 (3W)	" 14
15 (3X)	" 15
16 (4U)	" 16
17 (4V)	" 17
18 (4W)	" 18
19 (4X)	" 19 (IN-OUT)
20 (5U)	" 20 (4 word)
21 (5V)	" 21 (4 word)
22 (5W)	" 22 (4 word)
23 (5X)	" 23 (4 word, IN-OUT)
24 (6U)	MQ Register
25 (6V)	ID Register
26 (6W)	PN Register
27 (6X)	$20 \cdot 21 + \overline{20} \cdot AR$
28 (7U)	AR
29 (7V) $	*20•(INPUT REGISTER)
30 (7W)	$\overline{20} \cdot 21$
31 (7X)	20•21

D CODE	DESTINATION
00 (OU)	Line 0
01 (OV)	" 1
02 (OW)	" 2
03 (OX)	" 3
04 (1U)	" 4
05 (1V)	" 5
06 (1W)	" 6
07 (1X)	" 7
08 (2U)	" 8
09 (2V)	" 9
10 (2W)	" 10
11 (2X)	" 11
12 (3U)	" 12
13 (3V)	" 13
14 (3W)	" 14
15 (3X)	" 15
16 (4U)	" 16
17 (4V)	" 17
18 (4W)	" 18
19 (4X)	" 19 (IN-OUT)
20 (5U)	" 20 (4 word)
21 (5V)	" 21 (4 word)
22 (5W)	" 22 (4 word)
23 (5X)	" 23 (4 word, IN-OUT)
24 (6U)	MQ Register
25 (6V)	ID Register
26 (6W)	PN Register (PN_c)
27 (6X)	TEST (non-zero)
28 (7U)	AR (AR_c)
29 (7V)	Add to AR (AR_+)
30 (7W)	Add to PN (PN_+)
31 (7X)	"SPECIAL DEST. #31"

MQ Register	=	MULTIPLIER-QUOTIENT Register (2 word)
ID Register	=	MULTIPLICAND-DENOMINATOR Register (2 word)
PN Register	=	PRODUCT-NUMERATOR Register (2 word)
AR	=	ACCUMULATOR REGISTER (1 word)

$ means auxiliary equipment is required for normal operation

* can be used as source of ZEROS for clearing lines.

SPECIAL COMMANDS TABLE (DESTINATION - 31)

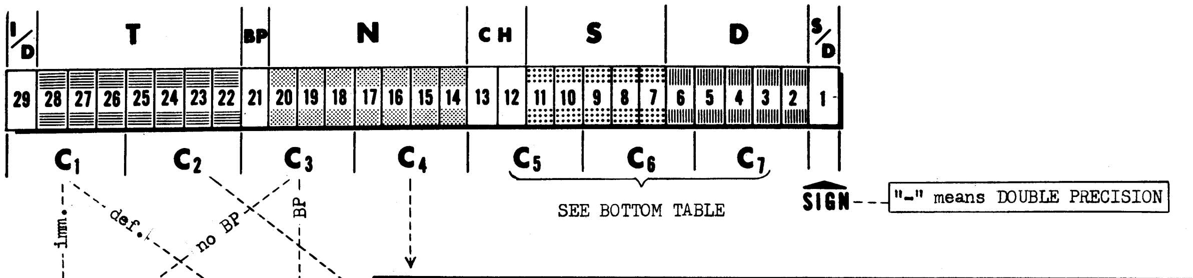
S CODE	CH CODE	HEX. CODE	SPECIAL COMMANDS (D CODE = 31)			OC's
00 (OU)		01z	SET "READY"	OUT	FAST	●●●● S
01 (OV)	0,1,2,3, $	03z*	MAGNETIC TAPE WRITE			0001
02 (OW)	$	05z	FAST PUNCH LEADER, etc.			0010
03 (OX)	$	07z	FAST PUNCH M19, etc.			0011
04 (1U)	0,1,2,3 $	09z*	MAGNETIC TAPE SEARCH, REV.	IN		0100
05 (1V)	0,1,2,3 $	0vz*	MAGNETIC TAPE SEARCH, FOR.			0101
06 (1W)		0xz	PHOTO TAPE REVERSE (Ø1)			0110 B
07 (1X)		0zz	PHOTO TAPE REVERSE (Ø2)			0111
08 (2U)		11z	TYPE AR } & punch if toggle	OUT	SLOW	1000 A
09 (2V)		13z	TYPE M19 } switch is set			1001
10 (2W)		15z	PUNCH M19			1010
11 (2X)	$	17z	CARD PUNCH M19			1011
12 (3U)		19z	TYPE IN	IN		1100 Q
13 (3V)	0,1,2,3 $	1vz*	MAGNETIC TAPE READ			1101
14 (3W)	$	1xz	CARD READ, etc.			1110
15 (3X)		1zz	PHOTO TAPE READ			1111 P

16 (4U)		21z	HALT	
17 (4V)	0	23z	RING BELL	
"	1	63z	RING BELL and <MAN. PUNCH> to TEST	
"	2 $	u3z	RING BELL and START INPUT REGISTER	
"	3 $	y3z	RING BELL and STOP INPUT REGISTER	
18 (4W)	$	25z	TRANSFER M20·ID to OUTPUT REGISTER	
19 (4X)	0 $	27z	START DA-1, etc.	
"	1 $	67z	STOP DA-1, etc.	
20 (5U)	0,1,2,3	29z*	SELECT COMMAND LINE (CH & S/D); RETURN EXIT	
21 (5V)	0,1,2,3	2vz*	SELECT COMMAND LINE (CH & S/D); MARK EXIT	
22 (5W)		2xz	SIGN OF AR to TEST (T1·AR → TEST)	
23 (5X)	0	2zz	CLEAR MQ, ID, PN, and IP F.F.	
"	3	yzz	PN·M2 → ID, and $PN \cdot \overline{M2}$ → PN	
24 (6U)		31z	MULTIPLY (ID X MQ plus PN_0 → PN)	T is rel.
25 (6V)	1	73z	DIVIDE (PN ÷ ID → MQ)	
26 (6W)	0,1	35z*	SHIFT MQ LEFT and ID RIGHT } increment AR if CH 0	
27 (6X)	0,1	37z*	NORMALIZE MQ	
28 (7U)	0	39z	"READY"	to TEST
"	1 $	79z	"READY IN"	
"	2 $	v9z	"READY OUT"	
"	3 $	z9z	DA-1 OFF	
29 (7V)		3vz	OVERFLOW	
30 (7W)	0,1,2,3 $	3xz*	MAGNETIC TAPE WRITE FILE CODE	
31 (7X)	0	3zz	NEXT COMMAND from AR	
"	1	7zz	TRANSFER NT to M18 (CN or $M18_0$ → M18)	
"	2		OR M18 + M20	

* means HEX. CODE FOR CH = 0 only

$ means auxiliary equipment is required for normal operation

DEC.	HEX.	BINARY
100 10 1	16 1	64 32 16 8 4 2 1
10^2 10^1 10^0	16^1 16^0	2^6 2^5 2^4 2^3 2^2 2^1 2^0
• • •	• •	• • • • • • •
• • 1	• 1	• • • • • • 1
• • 2	• 2	• • • • • 1 0
• • 3	• 3	• • • • • 1 1
• • 4	• 4	• • • • 1 0 0
• • 5	• 5	• • • • 1 0 1
• • 6	• 6	• • • • 1 1 0
• • 7	• 7	• • • • 1 1 1
• • 8	• 8	• • • 1 0 0 0
• • 9	• 9	• • • 1 0 0 1
• 1 0	• u	• • • 1 0 1 0
• 1 1	• v	• • • 1 0 1 1
• 1 2	• w	• • • 1 1 0 0
• 1 3	• x	• • • 1 1 0 1
• 1 4	• y	• • • 1 1 1 0
• 1 5	• z	• • • 1 1 1 1
• 1 6	1 0	• • 1 0 0 0 0
• 1 7	1 1	• • 1 0 0 0 1
• 1 8	1 2	• • 1 0 0 1 0
• 1 9	1 3	• • 1 0 0 1 1
• 2 0	1 4	• • 1 0 1 0 0
• 2 1	1 5	• • 1 0 1 0 1
• 2 2	1 6	• • 1 0 1 1 0
• 2 3	1 7	• • 1 0 1 1 1
• 2 4	1 8	• • 1 1 0 0 0
• 2 5	1 9	• • 1 1 0 0 1
• 2 6	1 u	• • 1 1 0 1 0
• 2 7	1 v	• • 1 1 0 1 1
• 2 8	1 w	• • 1 1 1 0 0
• 2 9	1 x	• • 1 1 1 0 1
• 3 0	1 y	• • 1 1 1 1 0
• 3 1	1 z	• • 1 1 1 1 1
• 3 2	2 0	• 1 0 0 0 0 0
• 3 3	2 1	• 1 0 0 0 0 1
• 3 4	2 2	• 1 0 0 0 1 0
• 3 5	2 3	• 1 0 0 0 1 1
• 3 6	2 4	• 1 0 0 1 0 0
• 3 7	2 5	• 1 0 0 1 0 1
• 3 8	2 6	• 1 0 0 1 1 0
• 3 9	2 7	• 1 0 0 1 1 1
• 4 0	2 8	• 1 0 1 0 0 0
• 4 1	2 9	• 1 0 1 0 0 1
• 4 2	2 u	• 1 0 1 0 1 0
• 4 3	2 v	• 1 0 1 0 1 1
• 4 4	2 w	• 1 0 1 1 0 0
• 4 5	2 x	• 1 0 1 1 0 1
• 4 6	2 y	• 1 0 1 1 1 0
• 4 7	2 z	• 1 0 1 1 1 1
• 4 8	3 0	• 1 1 0 0 0 0
• 4 9	3 1	• 1 1 0 0 0 1
• 5 0	3 2	• 1 1 0 0 1 0
• 5 1	3 3	• 1 1 0 0 1 1
• 5 2	3 4	• 1 1 0 1 0 0
• 5 3	3 5	• 1 1 0 1 0 1
• 5 4	3 6	• 1 1 0 1 1 0
• 5 5	3 7	• 1 1 0 1 1 1
• 5 6	3 8	• 1 1 1 0 0 0
• 5 7	3 9	• 1 1 1 0 0 1
• 5 8	3 u	• 1 1 1 0 1 0
• 5 9	3 v	• 1 1 1 0 1 1
• 6 0	3 w	• 1 1 1 1 0 0
• 6 1	3 x	• 1 1 1 1 0 1
• 6 2	3 y	• 1 1 1 1 1 0
• 6 3	3 z	• 1 1 1 1 1 1
• 6 4	4 0	1 0 0 0 0 0 0
• 6 5	4 1	1 0 0 0 0 0 1
• 6 6	4 2	1 0 0 0 0 1 0
• 6 7	4 3	1 0 0 0 0 1 1
• 6 8	4 4	1 0 0 0 1 0 0
• 6 9	4 5	1 0 0 0 1 0 1
• 7 0	4 6	1 0 0 0 1 1 0
• 7 1	4 7	1 0 0 0 1 1 1
• 7 2	4 8	1 0 0 1 0 0 0
• 7 3	4 9	1 0 0 1 0 0 1
• 7 4	4 u	1 0 0 1 0 1 0
• 7 5	4 v	1 0 0 1 0 1 1
• 7 6	4 w	1 0 0 1 1 0 0
• 7 7	4 x	1 0 0 1 1 0 1
• 7 8	4 y	1 0 0 1 1 1 0
• 7 9	4 z	1 0 0 1 1 1 1
• 8 0	5 0	1 0 1 0 0 0 0
• 8 1	5 1	1 0 1 0 0 0 1
• 8 2	5 2	1 0 1 0 0 1 0
• 8 3	5 3	1 0 1 0 0 1 1
• 8 4	5 4	1 0 1 0 1 0 0
• 8 5	5 5	1 0 1 0 1 0 1
• 8 6	5 6	1 0 1 0 1 1 0
• 8 7	5 7	1 0 1 0 1 1 1
• 8 8	5 8	1 0 1 1 0 0 0
• 8 9	5 9	1 0 1 1 0 0 1
• 9 0	5 u	1 0 1 1 0 1 0
• 9 1	5 v	1 0 1 1 0 1 1
• 9 2	5 w	1 0 1 1 1 0 0
• 9 3	5 x	1 0 1 1 1 0 1
• 9 4	5 y	1 0 1 1 1 1 0
• 9 5	5 z	1 0 1 1 1 1 1
• 9 6	6 0	1 1 0 0 0 0 0
• 9 7	6 1	1 1 0 0 0 0 1
• 9 8	6 2	1 1 0 0 0 1 0
• 9 9	6 3	1 1 0 0 0 1 1
1 0 0	6 4	1 1 0 0 1 0 0
1 0 1	6 5	1 1 0 0 1 0 1
1 0 2	6 6	1 1 0 0 1 1 0
1 0 3	6 7	1 1 0 0 1 1 1
1 0 4	6 8	1 1 0 1 0 0 0
1 0 5	6 9	1 1 0 1 0 0 1
1 0 6	6 u	1 1 0 1 0 1 0
1 0 7	6 v	1 1 0 1 0 1 1
1 0 8	6 w	1 1 0 1 1 0 0
1 0 9	6 x	1 1 0 1 1 0 1
1 1 0	6 y	1 1 0 1 1 1 0
1 1 1	6 z	1 1 0 1 1 1 1
1 1 2	7 0	1 1 1 0 0 0 0
1 1 3	7 1	1 1 1 0 0 0 1
1 1 4	7 2	1 1 1 0 0 1 0
1 1 5	7 3	1 1 1 0 0 1 1
1 1 6	7 4	1 1 1 0 1 0 0
1 1 7	7 5	1 1 1 0 1 0 1
1 1 8	7 6	1 1 1 0 1 1 0
1 1 9	7 7	1 1 1 0 1 1 1
1 2 0	7 8	1 1 1 1 0 0 0
1 2 1	7 9	1 1 1 1 0 0 1
1 2 2	7 u	1 1 1 1 0 1 0
1 2 3	7 v	1 1 1 1 0 1 1
1 2 4	7 w	1 1 1 1 1 0 0
1 2 5	7 x	1 1 1 1 1 0 1
1 2 6	7 y	1 1 1 1 1 1 0
1 2 7	7 z	1 1 1 1 1 1 1



		0	1	2	3	4	5	6	7	8	9	u	v	w	x	y	z
0	8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	9	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	u	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	v	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	w	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	x	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	y	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	z	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

T or N number

DESTINATIONS

SOURCE	Line Line 00 - 15	Line Line 16 - 19	Line MQ 20 - 24	ID 25	PN$_c$ 26	Test 27	AR$_c$ 28	AR+ 29	PN+ 30	31	SPECIAL COMMANDS*
00	000 - 00z	010 - 013	014 - 018	019	01u	01v	01w	01x	01y	01z	Set READY
01	020 - 02z	030 - 033	034 - 038	039	03u	03v	03w	03x	03y	03z	Mag. Write
02	040 - 04z	050 - 053	054 - 058	059	05u	05v	05w	05x	05y	05z	Fast Punch Ldr.
03	060 - 06z	070 - 073	074 - 078	079	07u	07v	07w	07x	07y	07z	Fast Punch M19
04	080 - 08z	090 - 093	094 - 098	099	09u	09v	09w	09x	09y	09z	Mag. Search rev.
05	0u0 - 0uz	0v0 - 0v3	0v4 - 0v8	0v9	0vu	0vv	0vw	0vx	0vy	0vz	Mag. Search fwd.
06	0w0 - 0wz	0x0 - 0x3	0x4 - 0x8	0x9	0xu	0xv	0xw	0xx	0xy	0xz	Ph Tape rev. Ø1
07	0y0 - 0yz	0z0 - 0z3	0z4 - 0z8	0z9	0zu	0zv	0zw	0zx	0zy	0zz	Ph Tape rev. Ø2
08	100 - 10z	110 - 113	114 - 118	119	11u	11v	11w	11x	11y	11z	Type AR
09	120 - 12z	130 - 133	134 - 138	139	13u	13v	13w	13x	13y	13z	Type M19
10	140 - 14z	150 - 153	154 - 158	159	15u	15v	15w	15x	15y	15z	Punch M19
11	160 - 16z	170 - 173	174 - 178	179	17u	17v	17w	17x	17y	17z	Punch Card M19
12	180 - 18z	190 - 193	194 - 198	199	19u	19v	19w	19x	19y	19z	Type In
13	1u0 - 1uz	1v0 - 1v3	1v4 - 1v8	1v9	1vu	1vv	1vw	1vx	1vy	1vz	Mag. Read
14	1w0 - 1wz	1x0 - 1x3	1x4 - 1x8	1x9	1xu	1xv	1xw	1xx	1xy	1xz	Card Read
15	1y0 - 1yz	1z0 - 1z3	1z4 - 1z8	1z9	1zu	1zv	1zw	1zx	1zy	1zz	Ph Tape Read
16	200 - 20z	210 - 213	214 - 218	219	21u	21v	21w	21x	21y	21z	Halt
17	220 - 22z	230 - 233	234 - 238	239	23u	23v	23w	23x	23y	23z	Ring Bell
18	240 - 24z	250 - 253	254 - 258	259	25u	25v	25w	25x	25y	25z	20•ID to OR
19	260 - 26z	270 - 273	274 - 278	279	27u	27v	27w	27x	27y	27z	Start DA-1
20	280 - 28z	290 - 293	294 - 298	299	29u	29v	29w	29x	29y	29z	Return
21	2u0 - 2uz	2v0 - 2v3	2v4 - 2v8	2v9	2vu	2vv	2vw	2vx	2vy	2vz	Mark
22	2w0 - 2wz	2x0 - 2x3	2x4 - 2x8	2x9	2xu	2xv	2xw	2xx	2xy	2xz	Test T1•AR
23	2y0 - 2yz	2z0 - 2z3	2z4 - 2z8	2z9	2zu	2zv	2zw	2zx	2zy	2zz	PG Clear
MQ 24	300 - 30z	310 - 313	314 - 318	319	31u	31v	31w	31x	31y	31z	Multiply
ID 25	320 - 32z	330 - 333	334 - 338	339	33u	33v	33w	33x	33y	73z	Divide (CH=1)
PN 26	340 - 34z	350 - 353	354 - 358	359	35u	35v	35w	35x	35y	35z	Shift
20•21 + $\overline{20}$•AR 27	360 - 36z	370 - 373	374 - 378	379	37u	37v	37w	37x	37y	37z	Normalize
AR 28	380 - 38z	390 - 393	394 - 398	399	39u	39v	39w	39x	39y	39z	Test READY
20•IR 29	3u0 - 3uz	3v0 - 3v3	3v4 - 3v8	3v9	3vu	3vv	3vw	3vx	3vy	3vz	Test OVERFLOW
$\overline{20}$•21 30	3w0 - 3wz	3x0 - 3x3	3x4 - 3x8	3x9	3xu	3xv	3xw	3xx	3xy	3xz	Mag. File Code
20•21 31	3y0 - 3yz	3z0 - 3z3	3z4 - 3z8	3z9	3zu	3zv	3zw	3zx	3zy	3zz	Next Comm. fm. AR

* SPECIAL COMMANDS: See "SPECIAL COMMANDS" table (page 3) for complete listing.

Note: The CHARACTERISTIC used in the above table is 0 (except for DIVIDE).

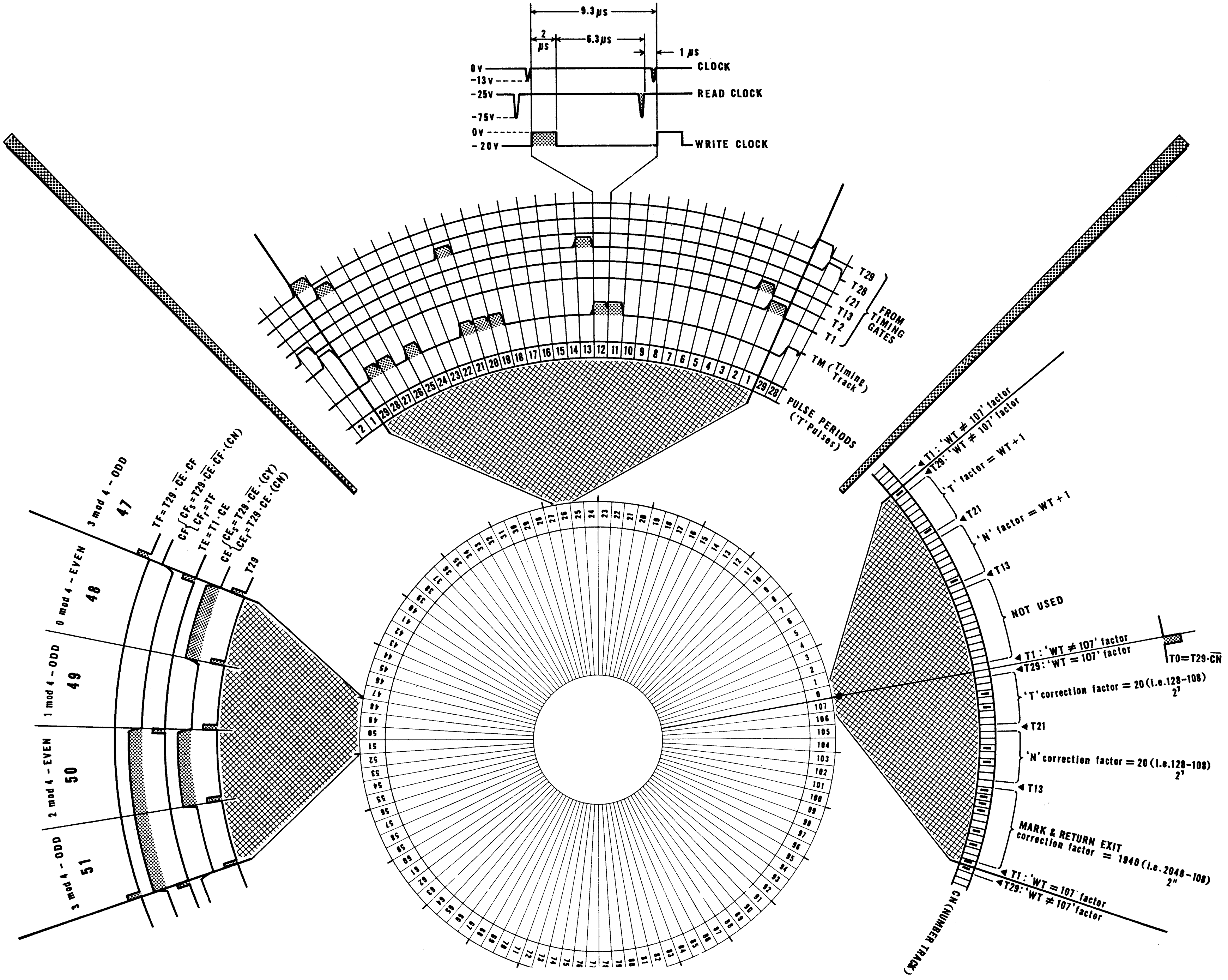
If CH = 1, add 4 to the left digit; if CH = 2, add 8; if CH = 3, add w

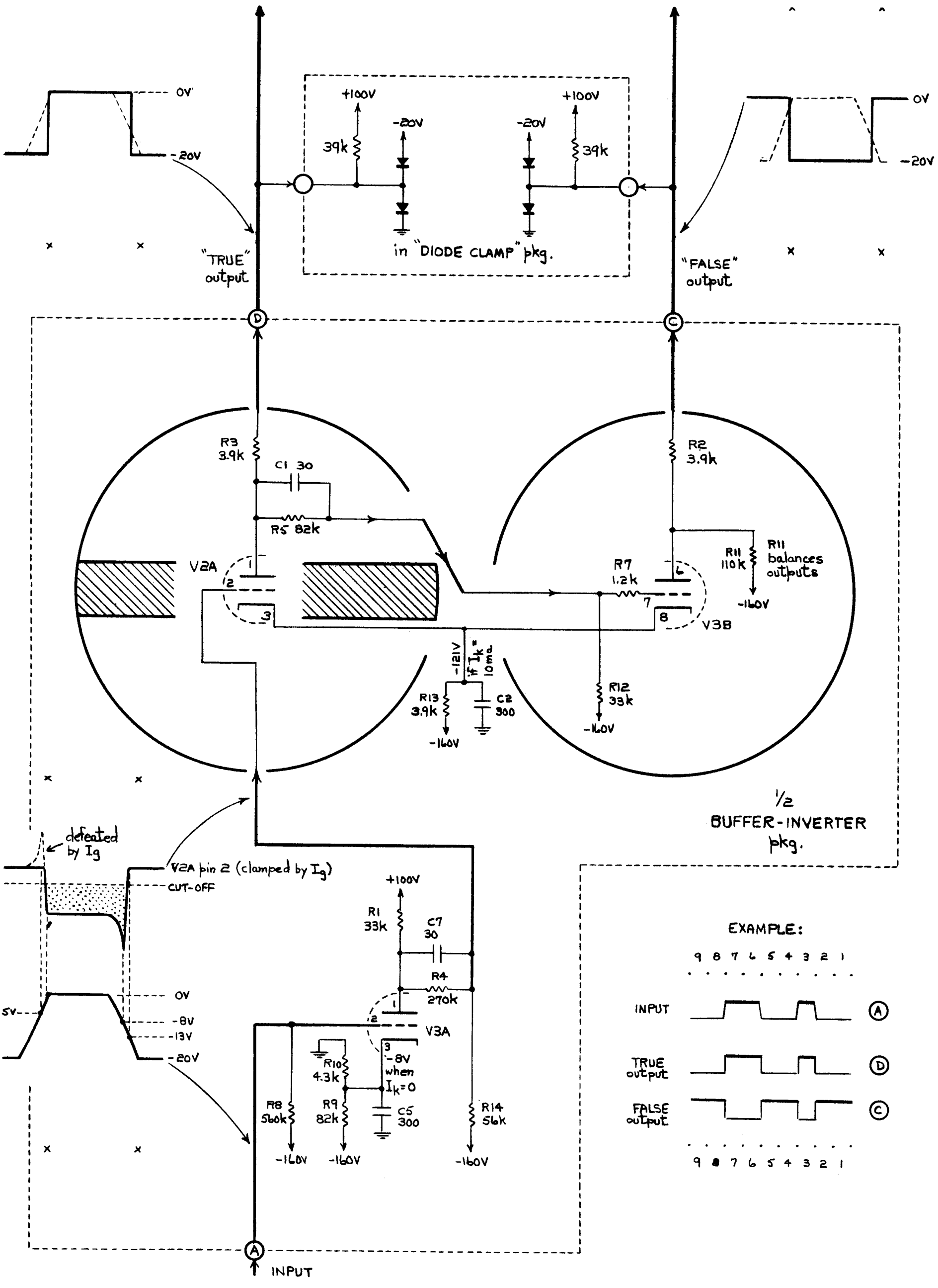
Examples: CH = 1, S = 28, D = 01, last three hex. digits = 781
CH = 3, S = 21, D = 29, last three hex. digits = yvx

COMMAND CONVERSION TABLE

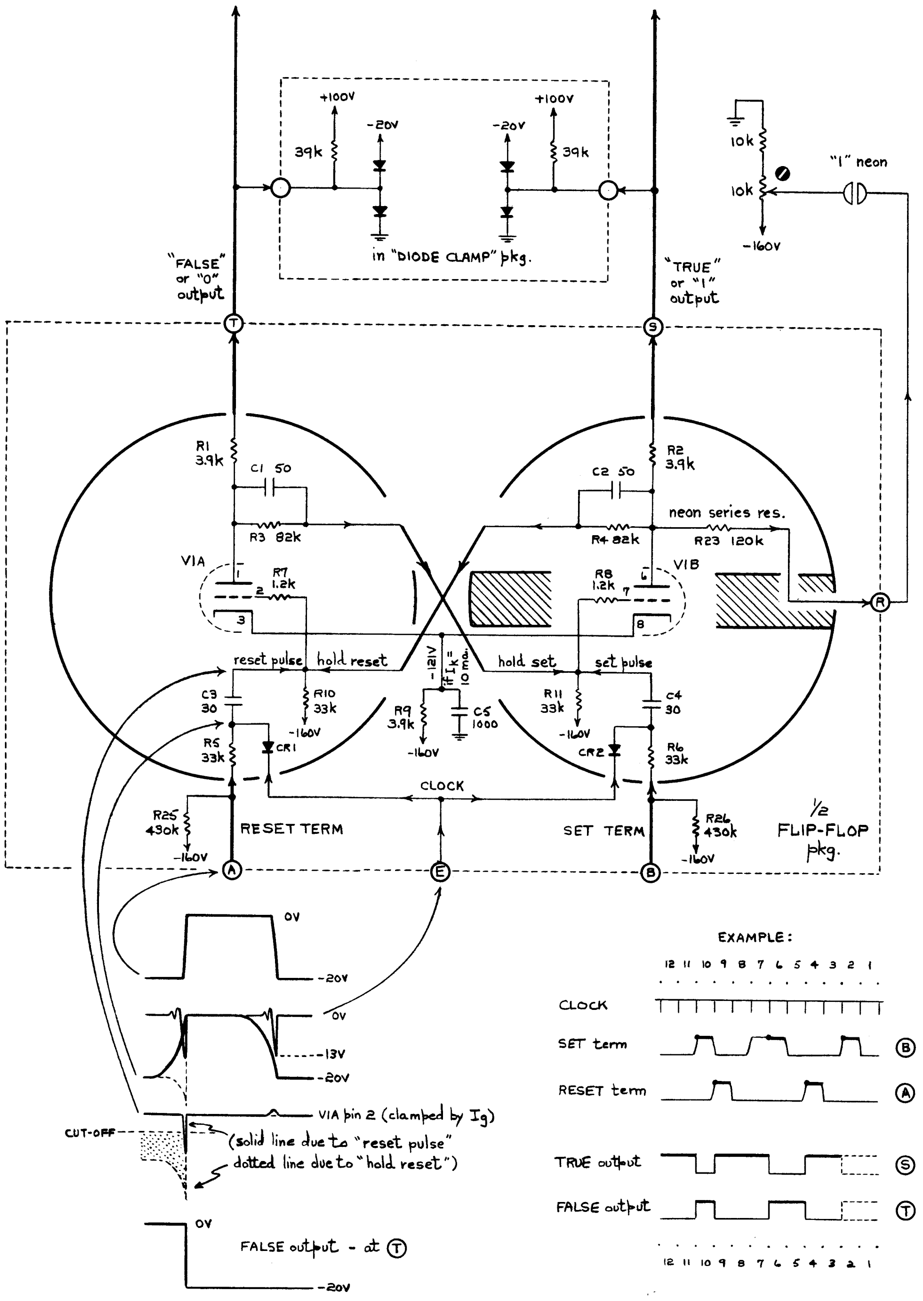
(Decimal to Hexadecimal)

G-15D

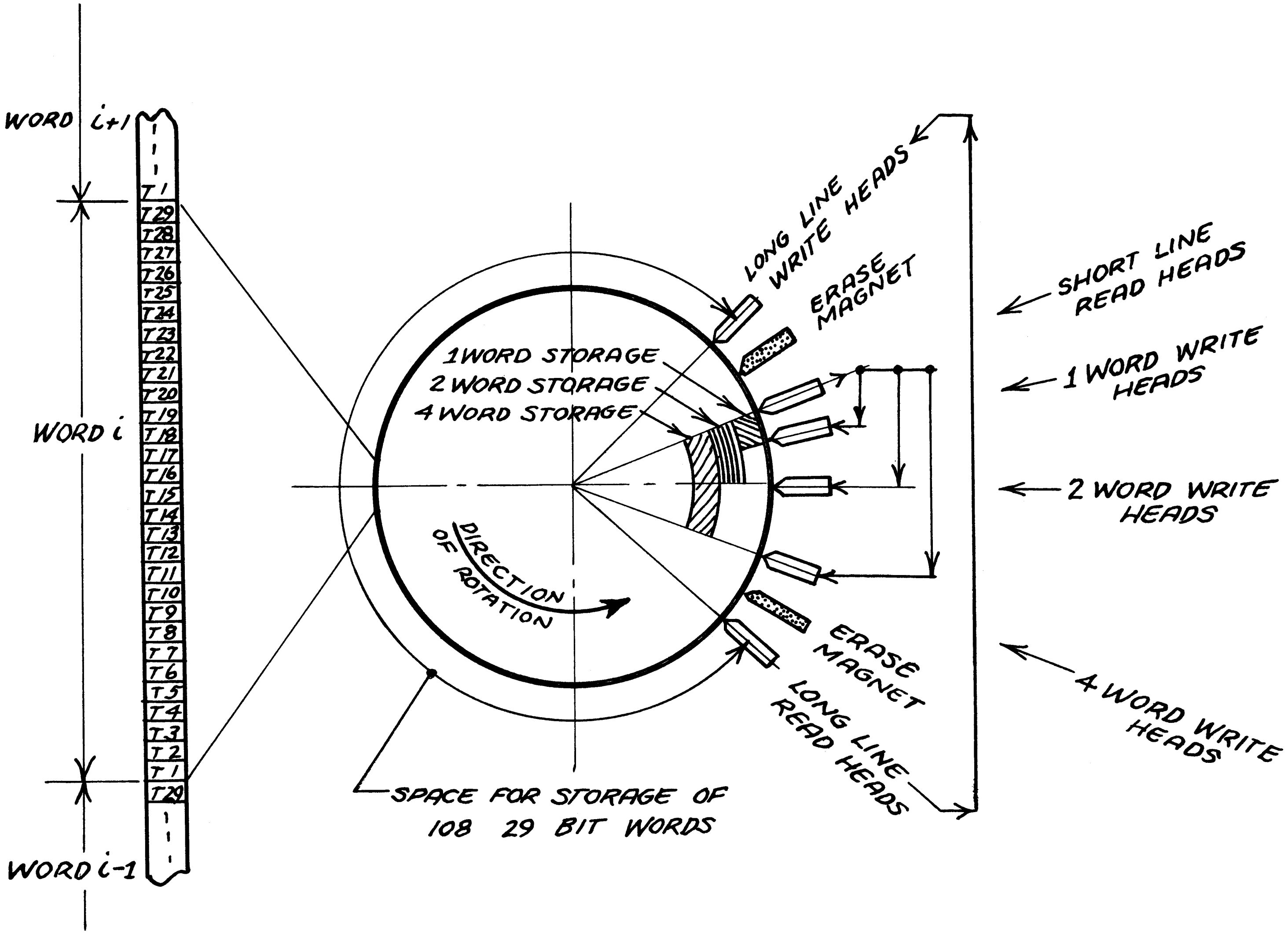

9.3 μs
2 μs
6.3 μs
1 μs
0v
-13v
CLOCK
-25v
-75v
READ CLOCK
0v
-20v
WRITE CLOCK
T29
T28
T21
T13
T2
T1
FROM TIMING GATES
TM (Timing Track)
PULSE PERIODS ('T' Pulses)
2 1 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 29 28
3 mod 4 – ODD
47
0 mod 4 – EVEN
48
1 mod 4 – ODD
49
2 mod 4 – EVEN
50
3 mod 4 – ODD
51
TF = T29 · $\overline{CE}$ · CF
CF { CF_s = T29 · $\overline{CE}$ · $\overline{CF}$ · (CN)
CF_r = TF
TE = T1 · CE
CE { CE_s = T29 · $\overline{CE}$ · (CY)
CE_r = T29 · CE · (CN)
T29
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
T1 : 'WT ≠ 107' factor
T29 : 'WT ≠ 107' factor
'T' factor = WT + 1
T21
'N' factor = WT + 1
T13
NOT USED
T1 : 'WT ≠ 107' factor
T29 : 'WT = 107' factor
T0 = T29 · $\overline{CN}$
'T' correction factor = 20 (i.e. 128 – 108) 2^7
T21
'N' correction factor = 20 (i.e. 128 – 108) 2^7
T13
MARK & RETURN EXIT correction factor = 1940 (i.e. 2048 – 108) 2"
T1 : 'WT = 107' factor
T29 : 'WT ≠ 107' factor
CN (NUMBER TRACK)



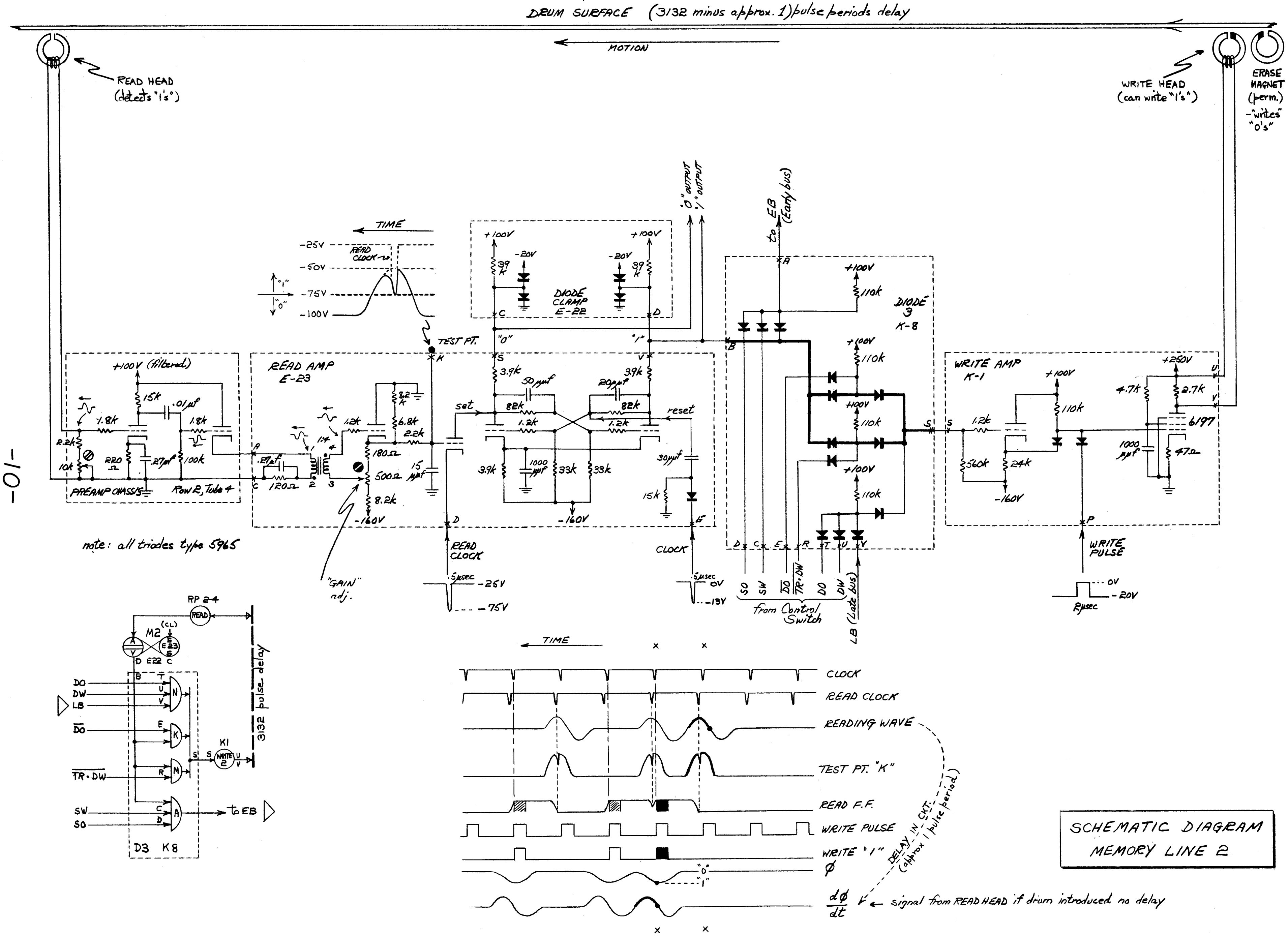
0V
-20V
"TRUE" output
+100V
39k
-20V
-20V
+100V
39k
in "DIODE CLAMP" pkg.
0V
-20V
"FALSE" output
D
C
R3 3.9k
C1 30
R5 82k
V2A
R2 3.9k
R11 110k
balances outputs
-160V
R7 1.2k
V3B
-121V
if Ik = 10 ma.
R13 3.9k
C2 300
-160V
R12 33k
-160V
1/2 BUFFER-INVERTER pkg.
defeated by Ig
V2A pin 2 (clamped by Ig)
CUT-OFF
-5V
0V
-8V
-13V
-20V
+100V
R1 33k
C7 30
R4 270k
V3A
R10 4.3k
-8V when Ik=0
R9 82k
C5 300
R8 560k
R14 56k
-160V
-160V
-160V
A
INPUT
EXAMPLE:
9 8 7 6 5 4 3 2 1
INPUT
A
TRUE output
D
FALSE output
C
9 8 7 6 5 4 3 2 1



+100V
39k
-20V
-20V
+100V
39k
in "DIODE CLAMP" pkg.
"FALSE" or "0" output
"TRUE" or "1" output
T
S
10k
10k
-160V
"1" neon
R1 3.9k
C1 50
R3 82k
V1A
R7 1.2k
R2 3.9k
C2 50
neon series res.
R4 82k
R23 120k
V1B
R8 1.2k
R
reset pulse
hold reset
hold set
set pulse
C3 30
R10 33k
-160V
-121V
if Ik = 10 ma.
R9 3.9k
-160V
C5 1000
R11 33k
-160V
C4 30
R5 33k
CR1
CR2
R6 33k
CLOCK
R25 430k
-160V
RESET TERM
SET TERM
R26 430k
-160V
1/2 FLIP-FLOP pkg.
A
E
B
0V
-20V
0V
-13V
-20V
V1A pin 2 (clamped by Ig)
CUT-OFF
(solid line due to "reset pulse"
dotted line due to "hold reset")
0V
FALSE output - at T
-20V
EXAMPLE:
12 11 10 9 8 7 6 5 4 3 2 1
CLOCK
SET term
B
RESET term
A
TRUE output
S
FALSE output
T
12 11 10 9 8 7 6 5 4 3 2 1



SECTION G-15 DRUM

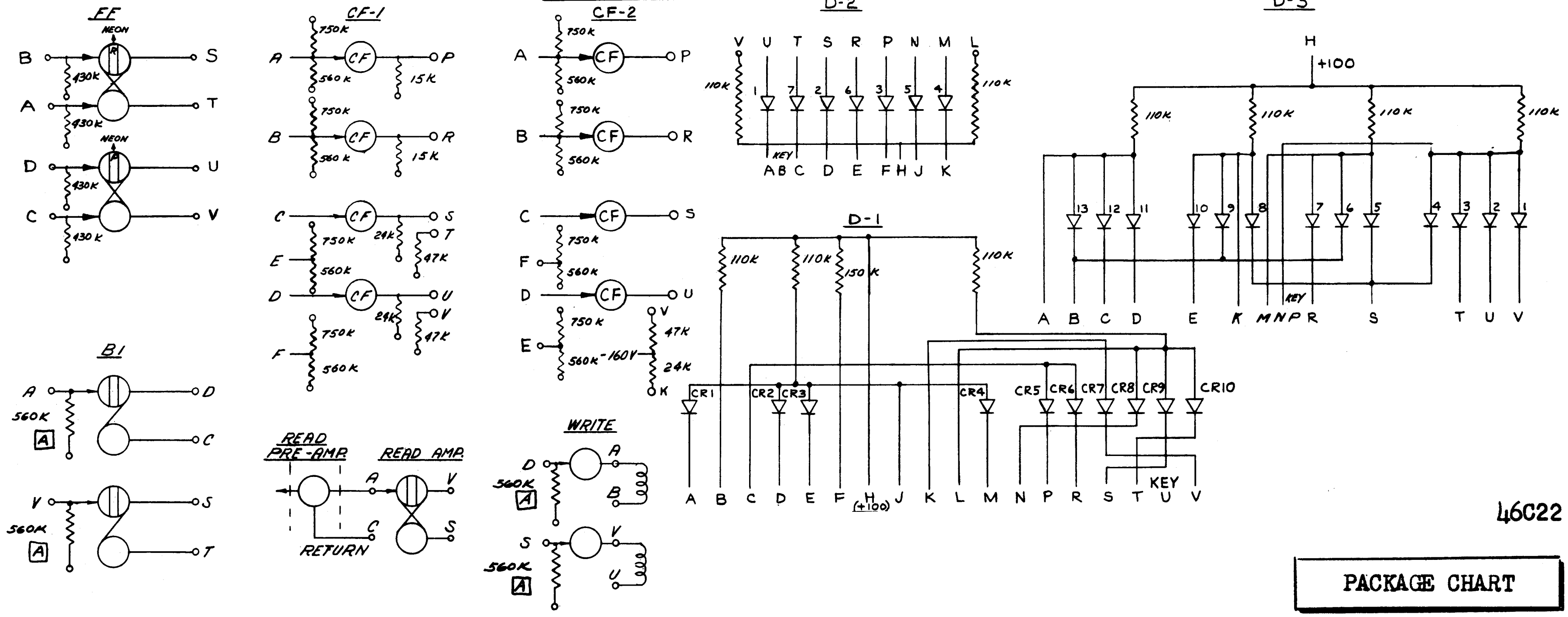


SCHEMATIC DIAGRAM
MEMORY LINE 2

PIN LETTERS

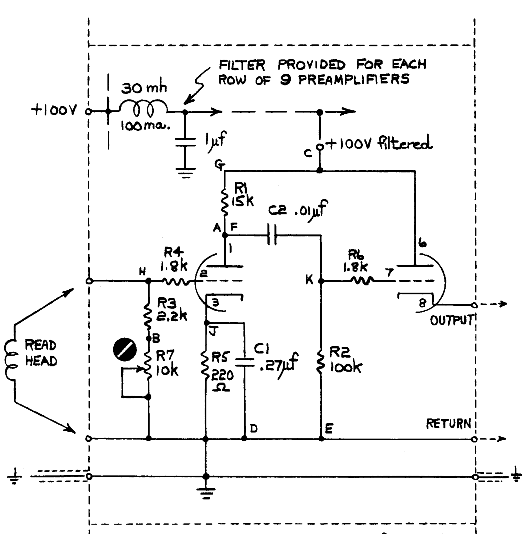
PART NO.	PACKAGE	PKG. SYMBOL	HANDLE COLOR	A	B	C	D	E	F	H	J	K	L	M	N	P	R	S	T	U	V
DL1083	FLIP FLOP	FF	YELLOW	INPUT 1B	INPUT 1A	INPUT 2B	INPUT 2A	CLOCK	OPEN	KEY	OVA	OPEN	-160V.	HTR @ 55V.	HTR @ 55V.	NEON 2	NEON 1	OUTPUT 1A	OUTPUT 1B	OUTPUT 2A	OUTPUT 2B
DL1084	BUFFER INVERTER	BI	BLUE	INPUT 1	OPEN	OUTPUT 1B	OUTPUT 1A	OPEN	OPEN	+100V.	OVA	OPEN	-160V.	HTR @55V.	HTR @55V.	KEY	OPEN	OUTPUT 2A	OUTPUT 2B	OPEN	INPUT 2
DL1082	CATHODE FOLLOWER #1	CF-1	RED	INPUT 1	INPUT 2	INPUT 3	INPUT 4	GRID DIVIDER	GRID DIVIDER	+100V	OVA	KEY	-160V.	HTR @55V.	HTR @55V	OUTPUT 1	OUTPUT 2	OUTPUT 3	47K TO B-	OUTPUT 4	47K TO B-
DL1081	READ AMP.	RA	GREEN	INPUT 1	SHIELD	RETURN	READ CLOCK	CLOCK	KEY	+100V	OVA	OPEN	-160V.	HTR @55V	HTR @55V	OPEN	OPEN	OUTPUT 1B	OPEN	OPEN	OUTPUT 1A
DL1089	WRITE AMP.	WA	BLACK	OUTPUT 1A	OUTPUT 1B	SHIELD	INPUT 1	KEY	+250V	+100V	OPEN	OVC	-160V.	HTR @55V	HTR @55V	WRITE PULSE	OPEN	INPUT 2	SHIELD	OUTPUT 2B	OUTPUT 2A
DL1088	DIODE CLAMP	DC		CL_1	CL_2	CL_3	CL_4	CL_5	CL_6	+100V	OVA	-20V	°JUMPER°		KEY	CL_7	CL_8	CL_9	CL_{10}	CL_{11}	CL_{12}
DL1163	DIODE 1	D1		C_1	110 K TO B+	$P_{5,6}$	C_2	C_3	150K TO B+	+100V	$P_{1,2,3,4.}$	P_7	$P_{8,9,10}$	C_4	C_8	C_5	C_6	C_9	C_{10}	KEY	C_7
DL1086	DIODE 2	D2		C_1	KEY	C_7	C_2	C_6	C_3	+100V	C_5	C_4	110K TO B+	P_4	P_5	P_3	P_6	P_2	P_7	P_1	110K TO B+
DL1085	DIODE 3	D3		$P_{11,12,13.}$	$C_{6,9,13.}$	C_{12}	C_{11}	C_{10}	OPEN	+100V	OPEN	$P_{8,9,10.}$	OPEN	$P_{5,6,7.}$	$P_{1,2,3,4.}$	KEY	C_7	$C_{4,5,8.}$	C_3	C_2	C_1
DL1169	CLOCK CLAMP	CC		OPEN	OPEN	OPEN	OPEN	CLOCK	OPEN	OPEN	OVA	-20V	-13V	OPEN	OPEN	WRITE PULSE	KEY	SP	LEVEL @-70V.	RC RETURN	READ PULSE (CLAMP)
DL1115	CATHODE FOLLOWER #2	CF-2	NATURAL	INPUT 1	INPUT 2	INPUT 3	INPUT 4	GRID DIVIDER	GRID DIVIDER	+100V	OVA	24K TO B-	-160V	HTR @55V.	HTR @55V.	OUTPUT 1	OUTPUT 2	OUTPUT 3	KEY	OUTPUT 4	47K TO B-

LOGIC SYMBOLS



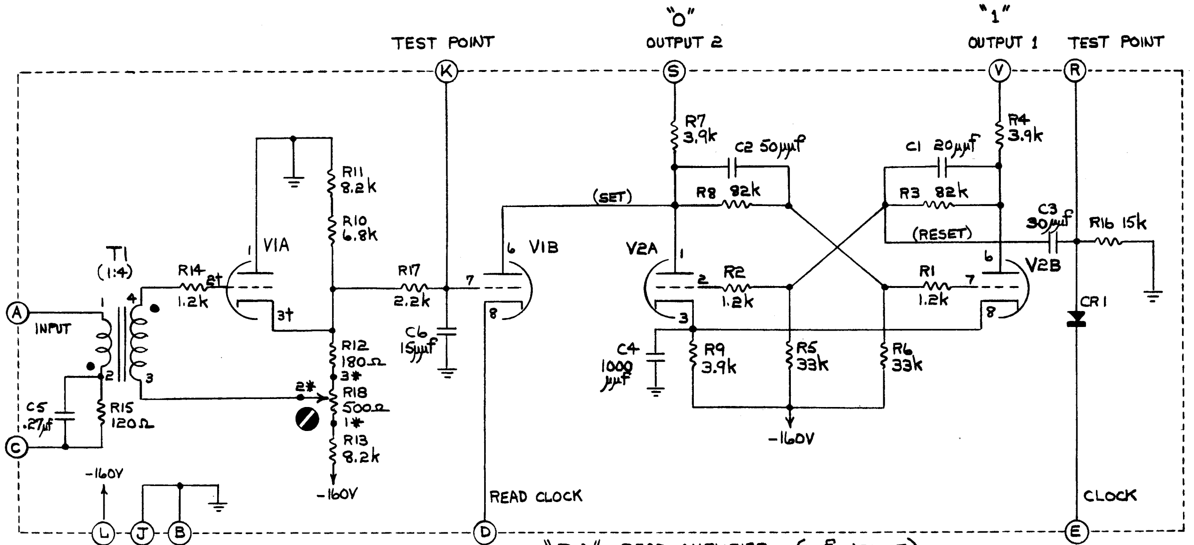
46C22

PACKAGE CHART



READ PREAMP. (CHANNEL) (ref: 3B111)

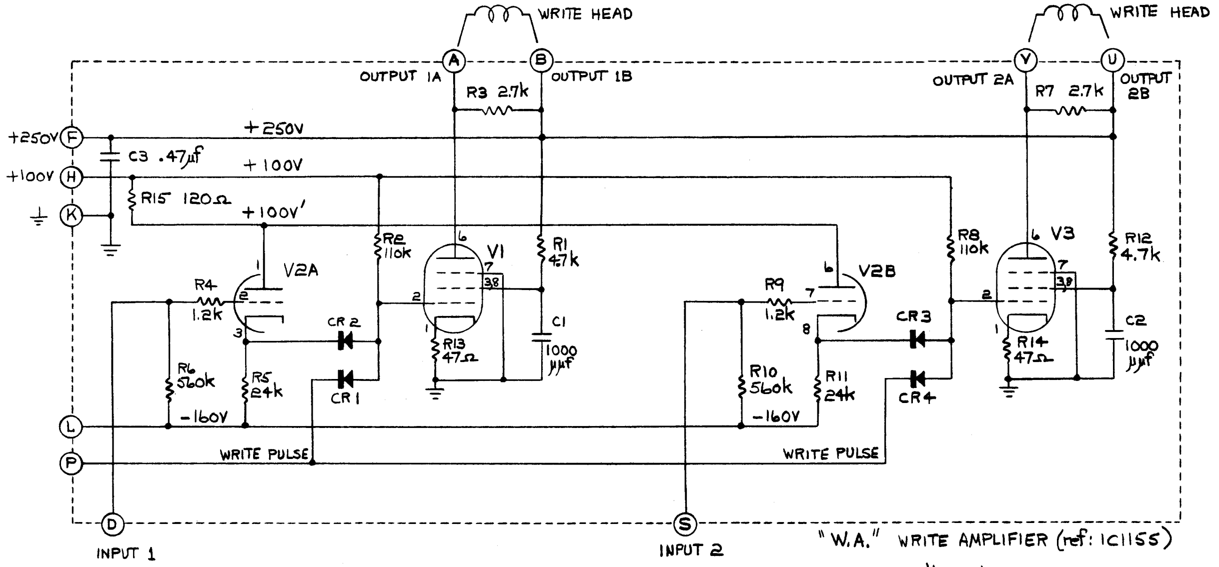
(LOCATED IN CHASSIS NEXT TO DRUM)



"R.A" READ AMPLIFIER (ref: 1C1007)
Handle color: GREEN

t* NOTES: with V1A out of socket and no signal on T1, potentials at pts. marked * or t are as follows:

1* (jct. R13+R18) = −105.06V
3* (jct. R18+R12) = −101.71V
2* or 2t (pot arm [R18], or V1A pin 2) = −105.06V < E < −101.71V
3t (V1A cathode) = −100.50
V1A, grid to cathode = −4.56 < E < −1.21V
Considering E_{p-k} of V1A = 100.50V, cut-off E_{g-k} = approx −4V



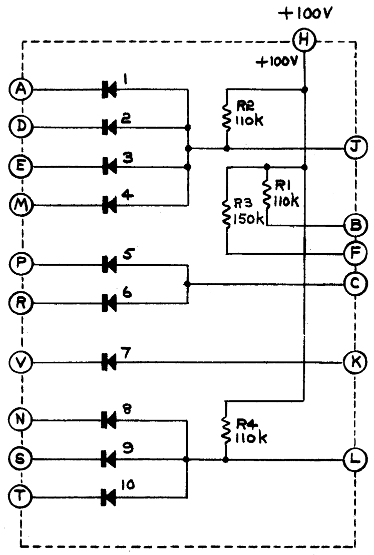
"W.A." WRITE AMPLIFIER (ref: 1C1155)
Handle color: BLACK

Notes: 1) all triodes type 5965
2) all pentodes type 6197
3) 6.3 VAC @ −55V is connected to Ⓜ and Ⓝ of each package.

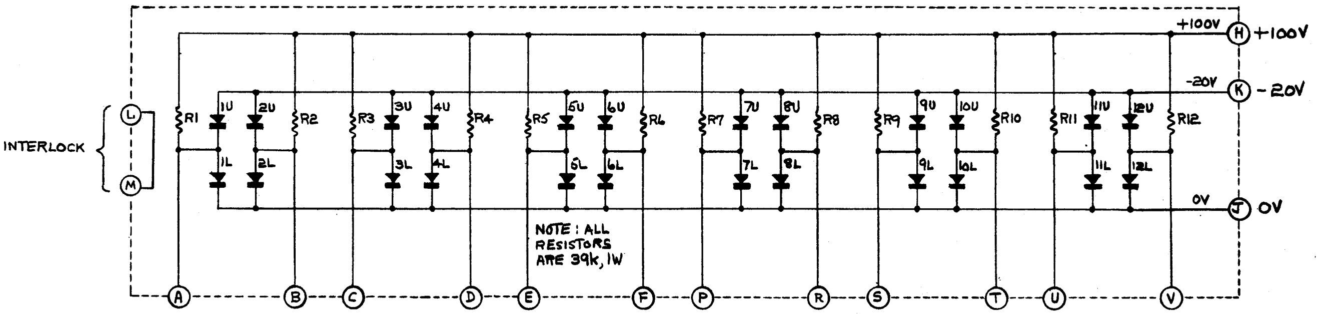
(ref. also 46C22)

READ-WRITE PACKAGES
& READ PREAMP.

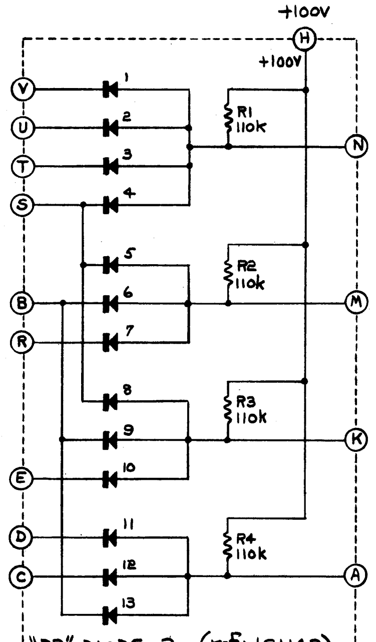
R.A., W.A.,



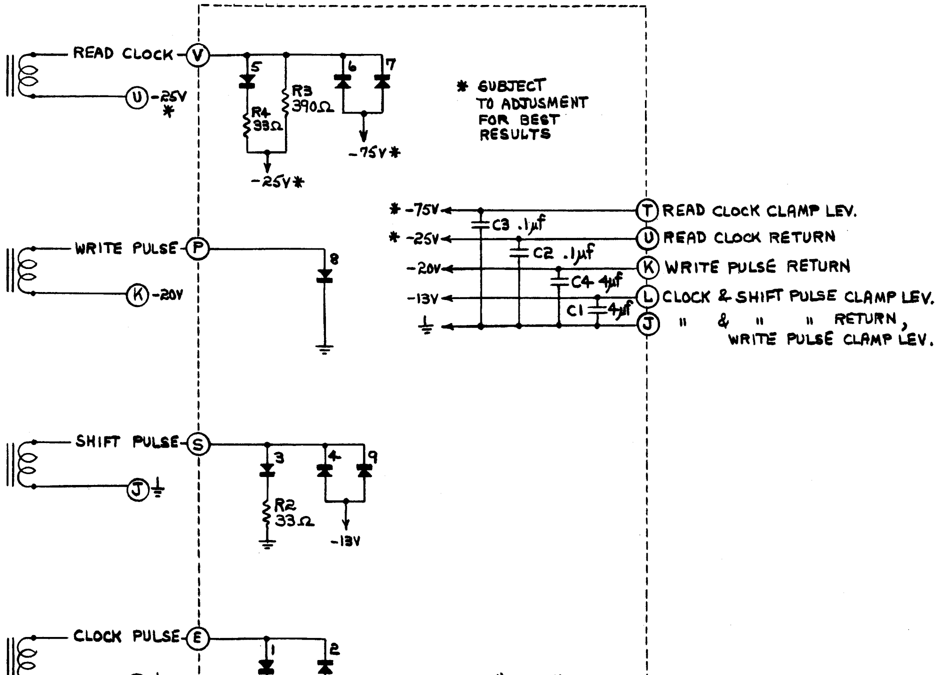
"D1" DIODE 1 (ref: 1C964)



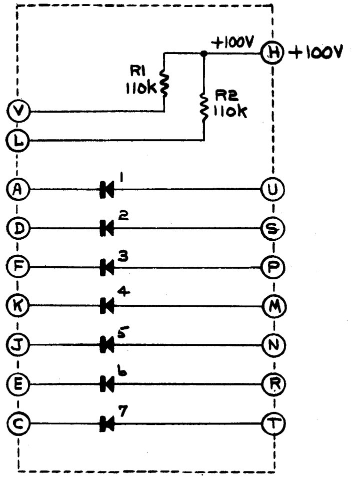
"D.C." DIODE CLAMP (ref: 1C993)
(for plate ckts of BUFFER INVERTERS and FLIP-FLOPS)



"D3" DIODE 3 (ref: 1C1142)



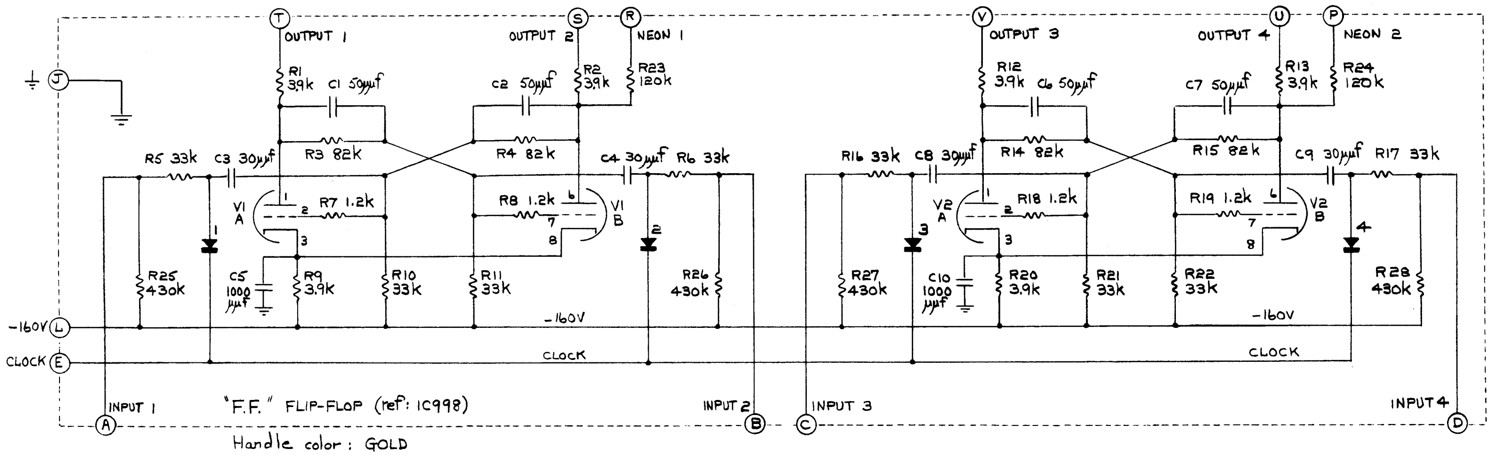
"C.C."
CLOCK CLAMP
(ref: 1C1009)
(For clock ckt. termination)



"D2" DIODE 2
(ref: 1C991)

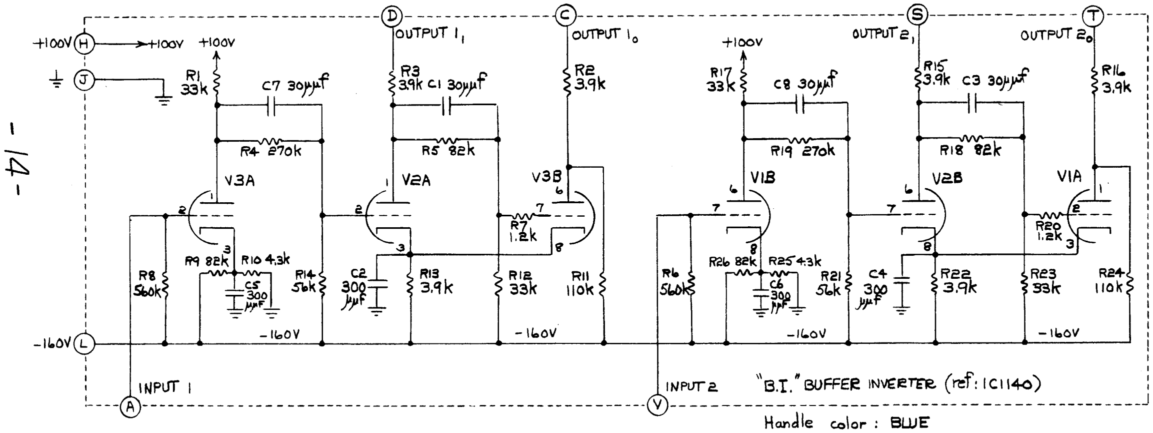
(ref. also 46C22)

DIODE PACKAGES (CARDS)
(For "and gate", "or gate" and clamping applications)



"F.F." FLIP-FLOP (ref: 1C998)

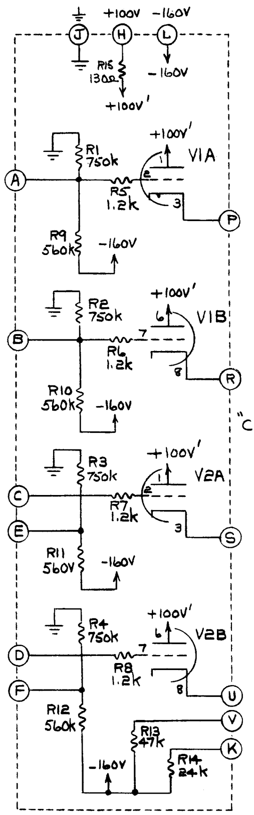
Handle color : GOLD



"B.I." BUFFER INVERTER (ref: 1C1140)

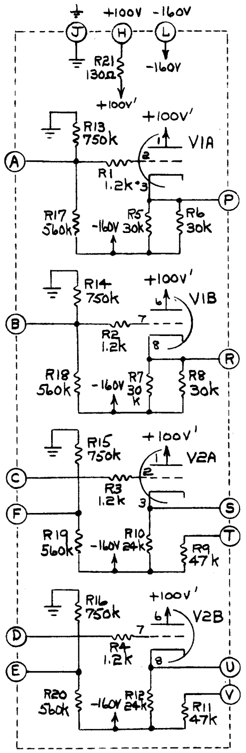
Handle color : BLUE

Notes: 1) All tubes type 5965
2) Buffer-inverter used for "HC" is modified as follows: C8 has been disconnected from V2B pin 7 and connected to ground to reduce high frequency response.
3) 6.3 VAC @ -55V is connected to Ⓜ and Ⓝ of each package



"CF2" CATHODE FOLLOWER #2 (ref: 1C1107)

Handle color: SILVER

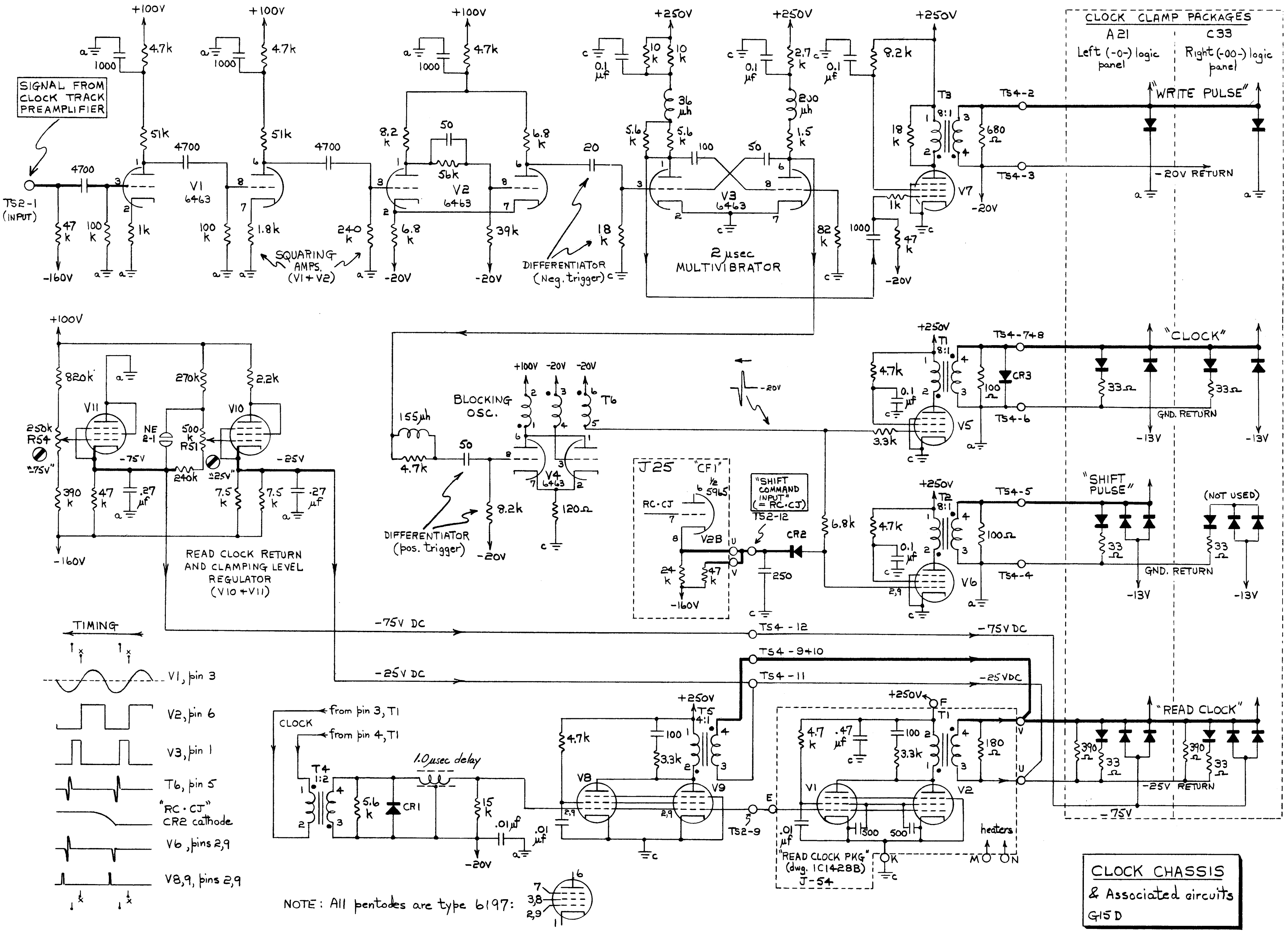


"CF1" CATHODE FOLLOWER #1 (ref: 1C1144)

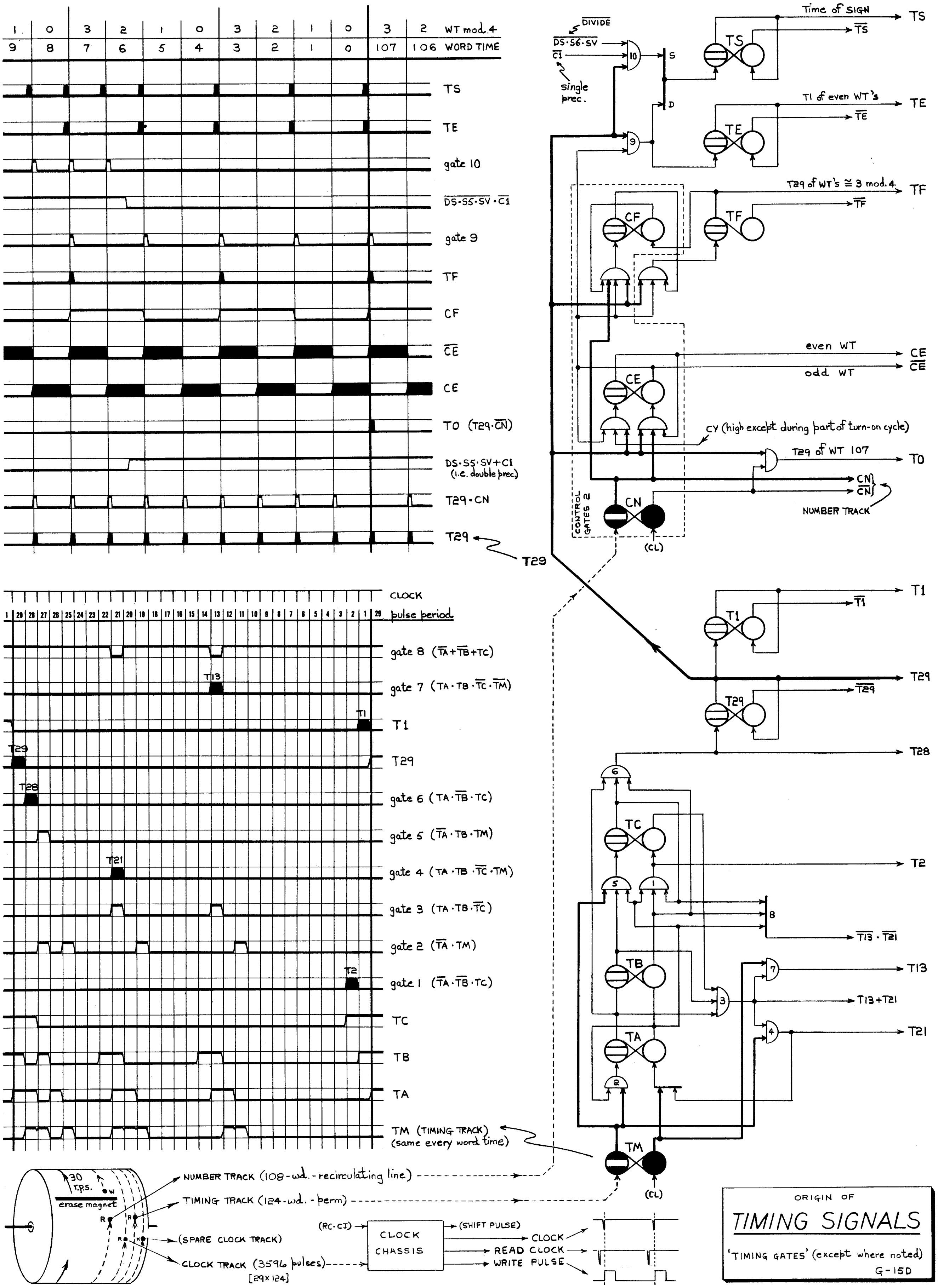
Handle color: RED

(ref. also 46C22)

TUBE PACKAGES
FF, BI, CF1, & CF2.



CLOCK CHASSIS
& Associated circuits
G15D

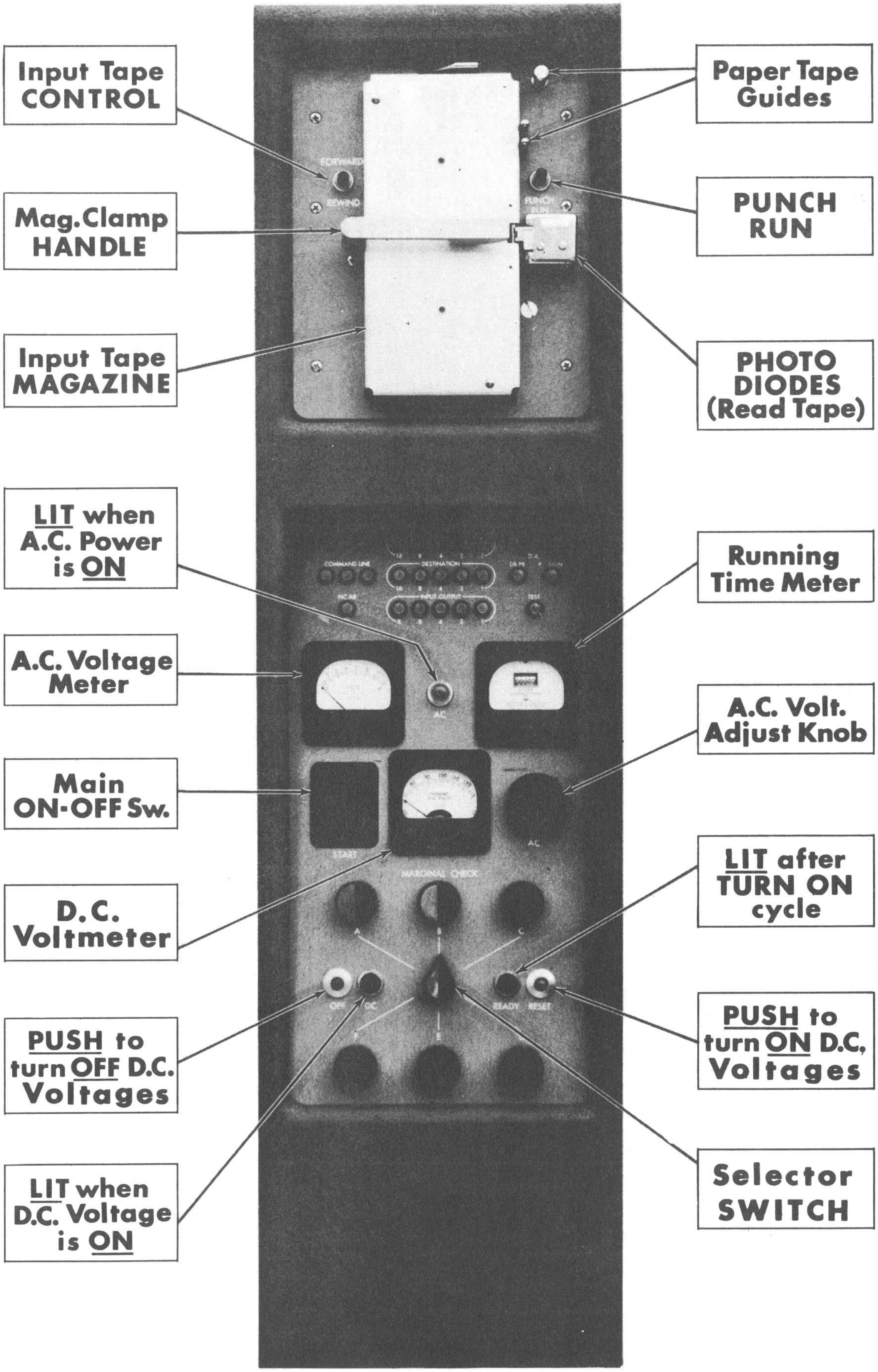


ORIGIN OF

TIMING SIGNALS

'TIMING GATES' (except where noted)

G-15D



Input Tape CONTROL

Mag.Clamp HANDLE

Input Tape MAGAZINE

LIT when A.C. Power is ON

A.C. Voltage Meter

Main ON-OFF Sw.

D. C. Voltmeter

PUSH to turn OFF D.C. Voltages

LIT when D.C. Voltage is ON

Paper Tape Guides

PUNCH RUN

PHOTO DIODES (Read Tape)

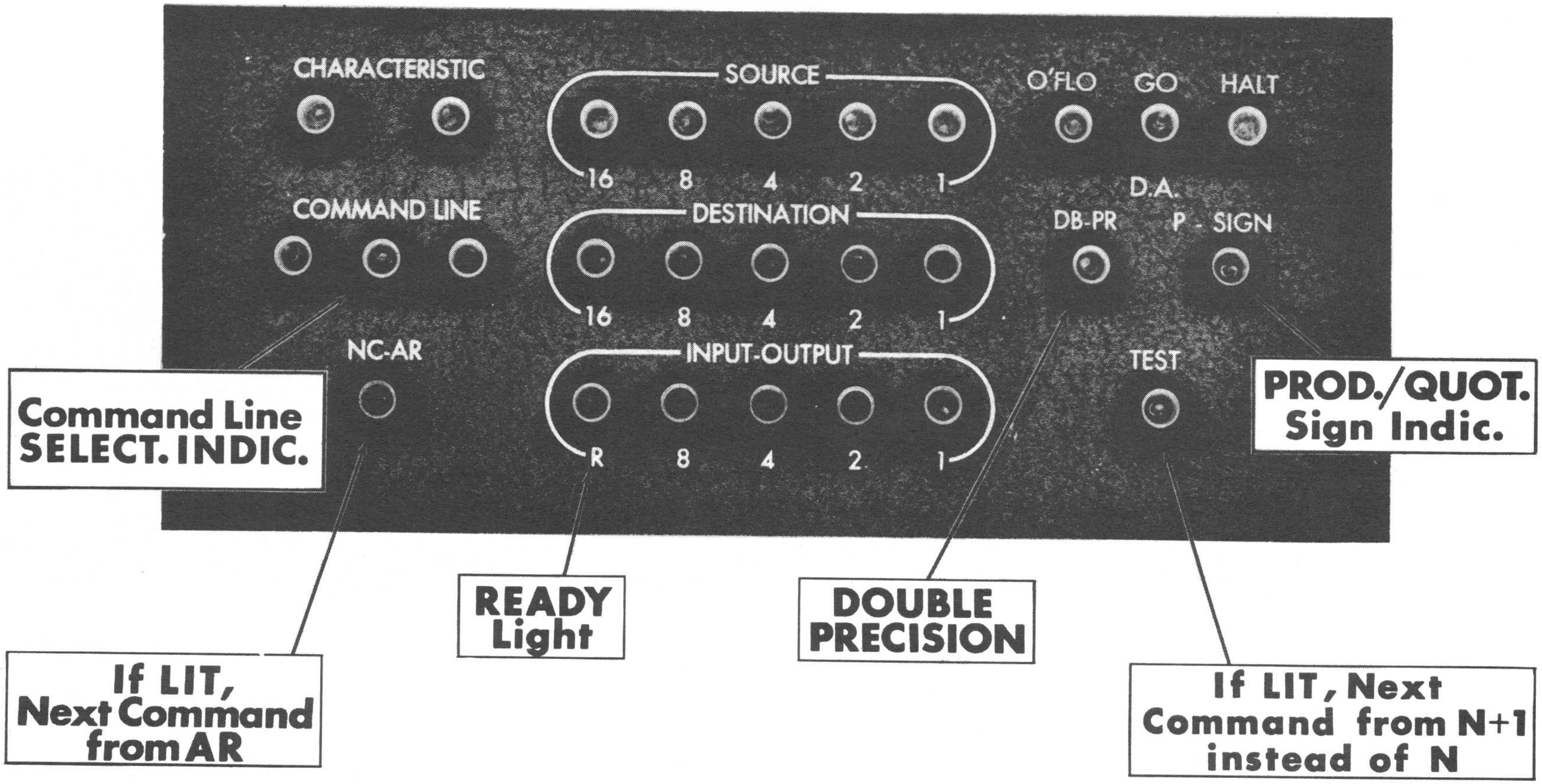
Running Time Meter

A.C. Volt. Adjust Knob

LIT after TURN ON cycle

PUSH to turn ON D.C. Voltages

Selector SWITCH



Command Line SELECT. INDIC.

PROD./QUOT. Sign Indic.

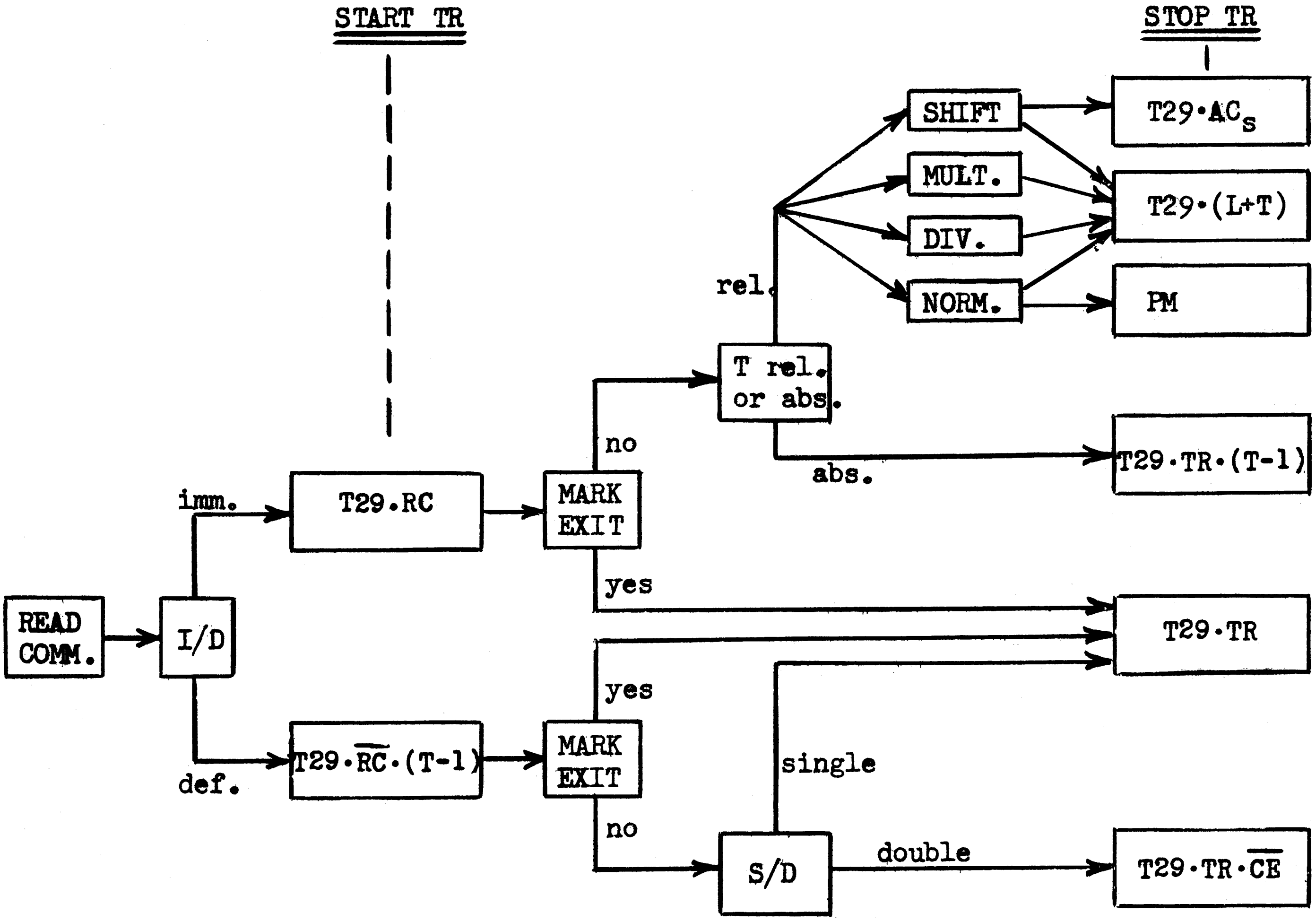
If LIT, Next Command from AR

READY Light

DOUBLE PRECISION

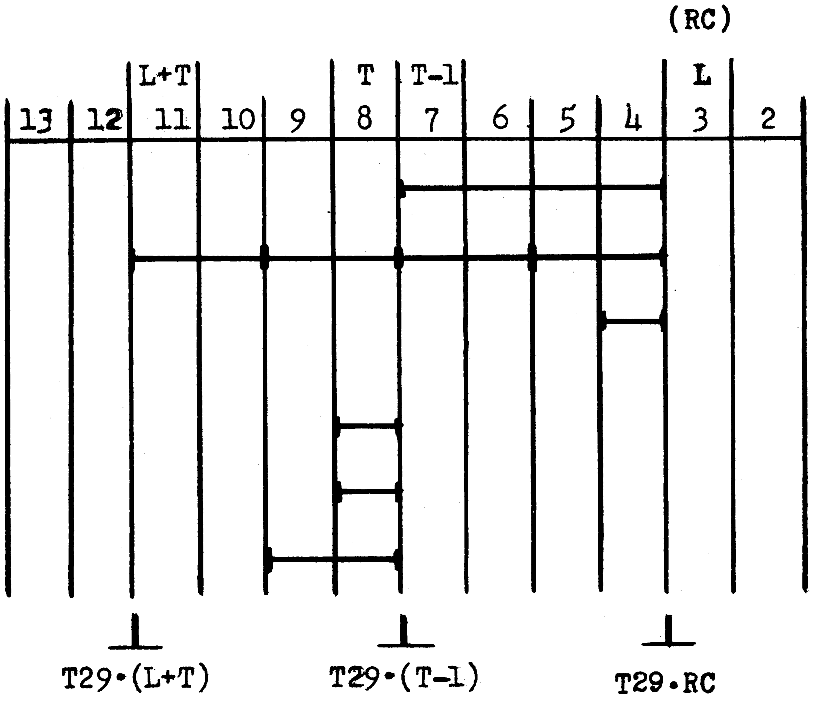
If LIT, Next Command from N+1 instead of N

TRANSFER TIMING



$\overline{CE}$ means ODD word

EXAMPLES:



WT_{RC} = L = 3, T = 8

Imm., $\overline{\text{MARK EX.}}$, abs.

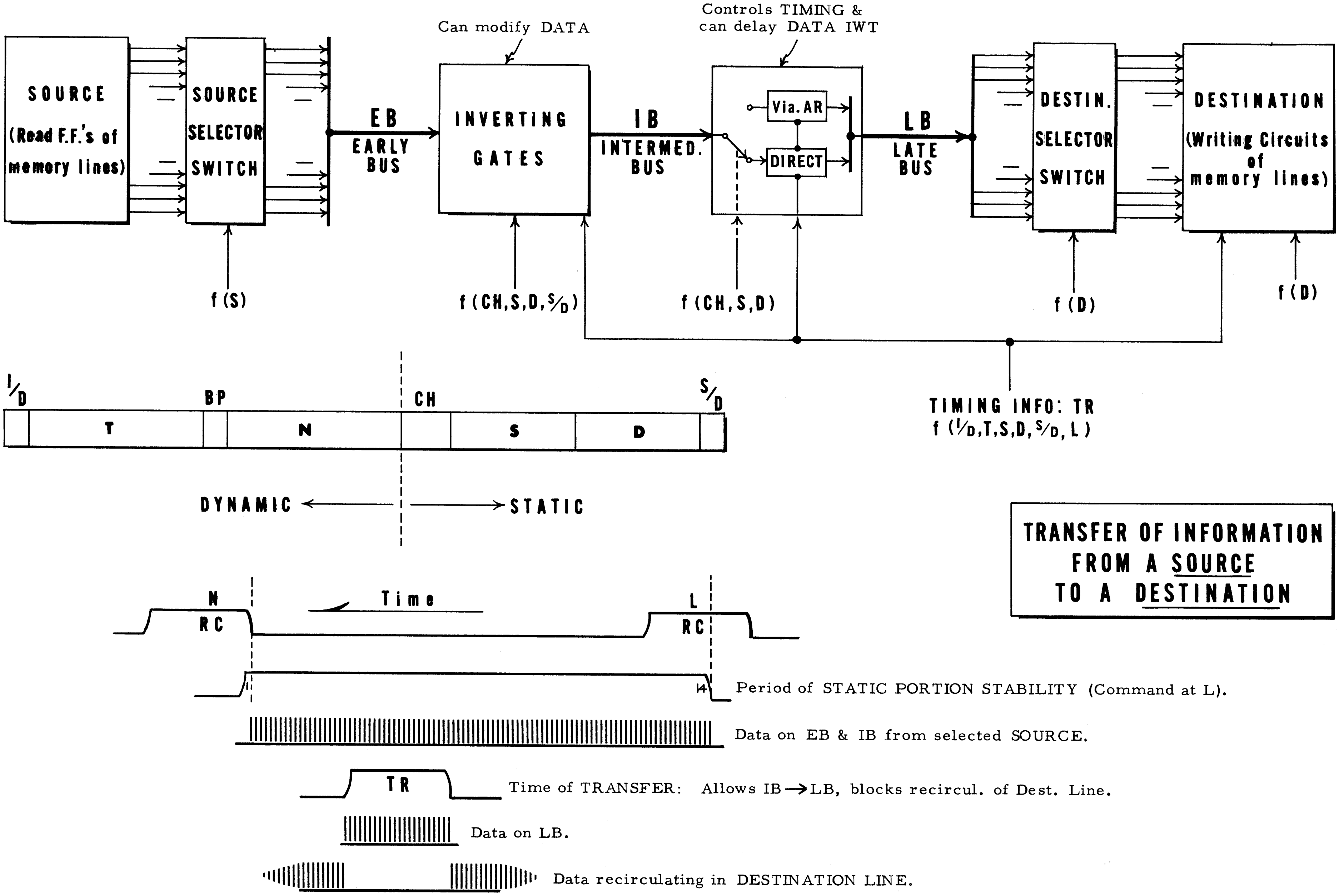
Imm., $\overline{\text{MARK EX.}}$, rel.

Imm., MARK EX.

Def., MARK EX.

Def., $\overline{\text{MARK EX.}}$, single

Def., $\overline{\text{MARK EX.}}$, double



Period of STATIC PORTION STABILITY (Command at L).

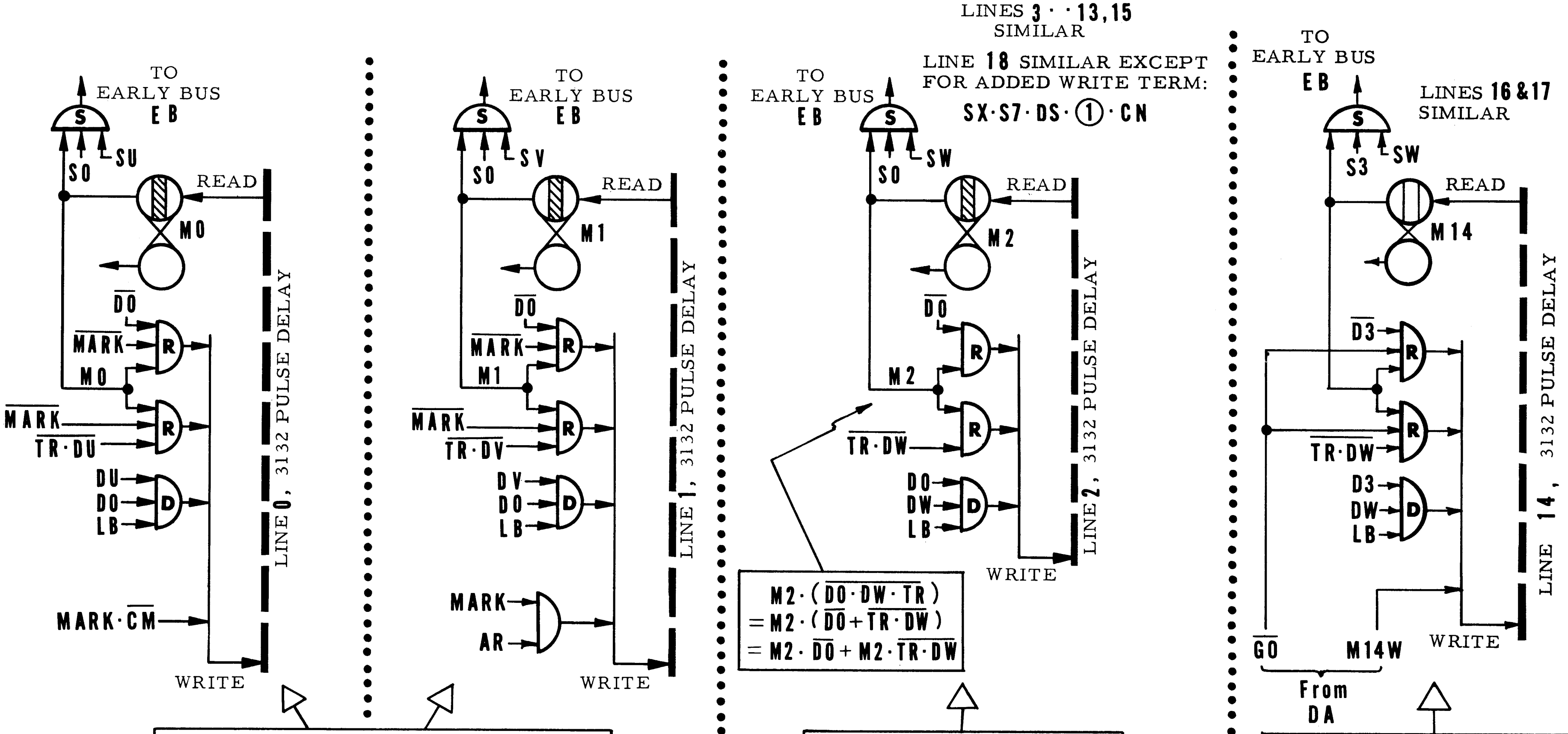
Data on EB & IB from selected SOURCE.

Time of TRANSFER: Allows IB → LB, blocks recircul. of Dest. Line.

Data on LB.

Data recirculating in DESTINATION LINE.

TRANSFER OF INFORMATION FROM A SOURCE TO A DESTINATION



LINES 3 · · 13,15
SIMILAR

LINE 18 SIMILAR EXCEPT
FOR ADDED WRITE TERM:
SX·S7·DS·①·CN

LINES 16 & 17
SIMILAR

LINES 0 & 1 RECIRCULATE AT ALL TIMES EXCEPT:

1. At time of transfer when the line is a DESTINATION.

2. During WORD TIME 107, when the **MARK** operation is being executed.

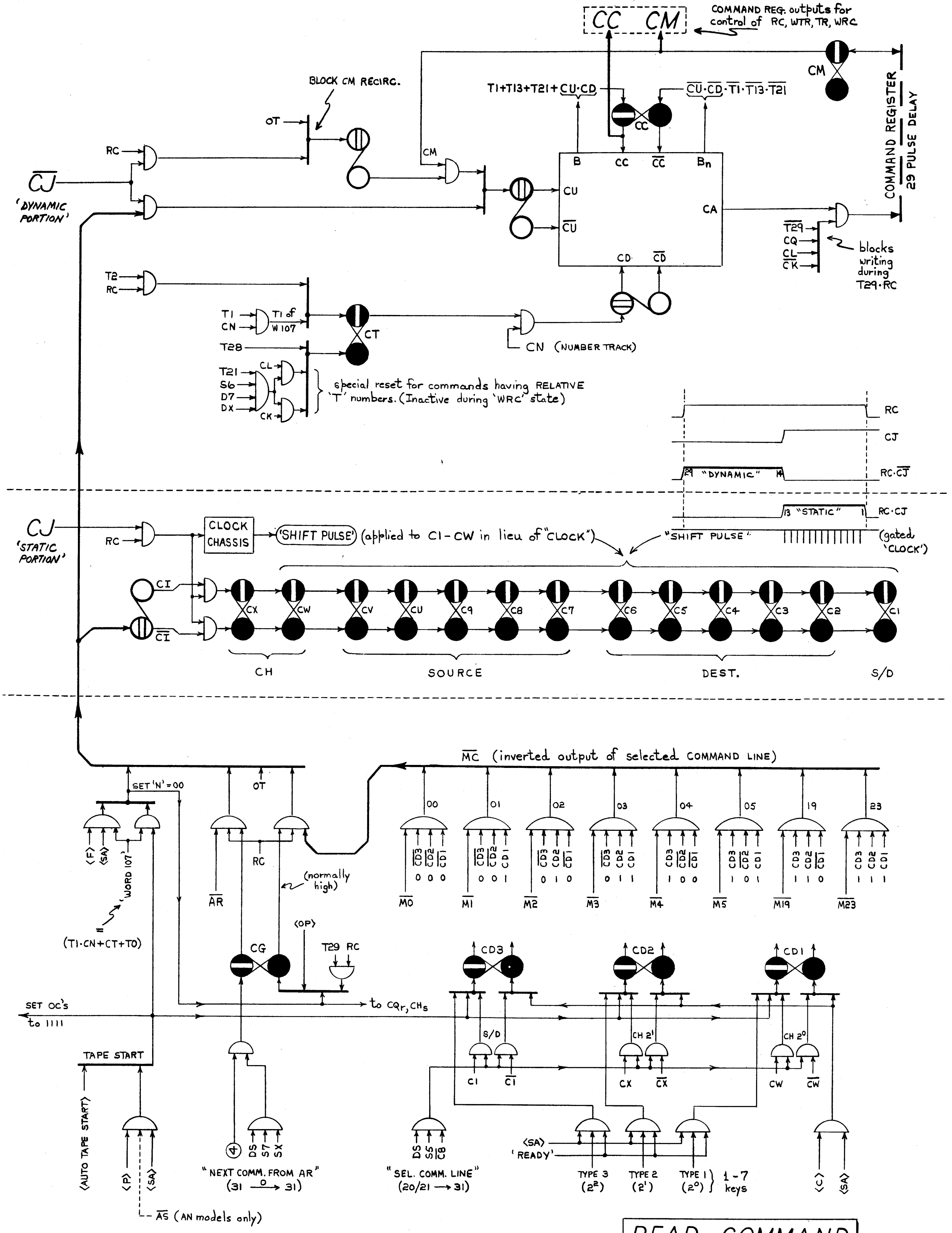
LINES 2, 3, · · · 13, 15 & 18 RECIRCULATE AT ALL TIMES EXCEPT

1. At time of TRANSFER when the line is a DESTINATION

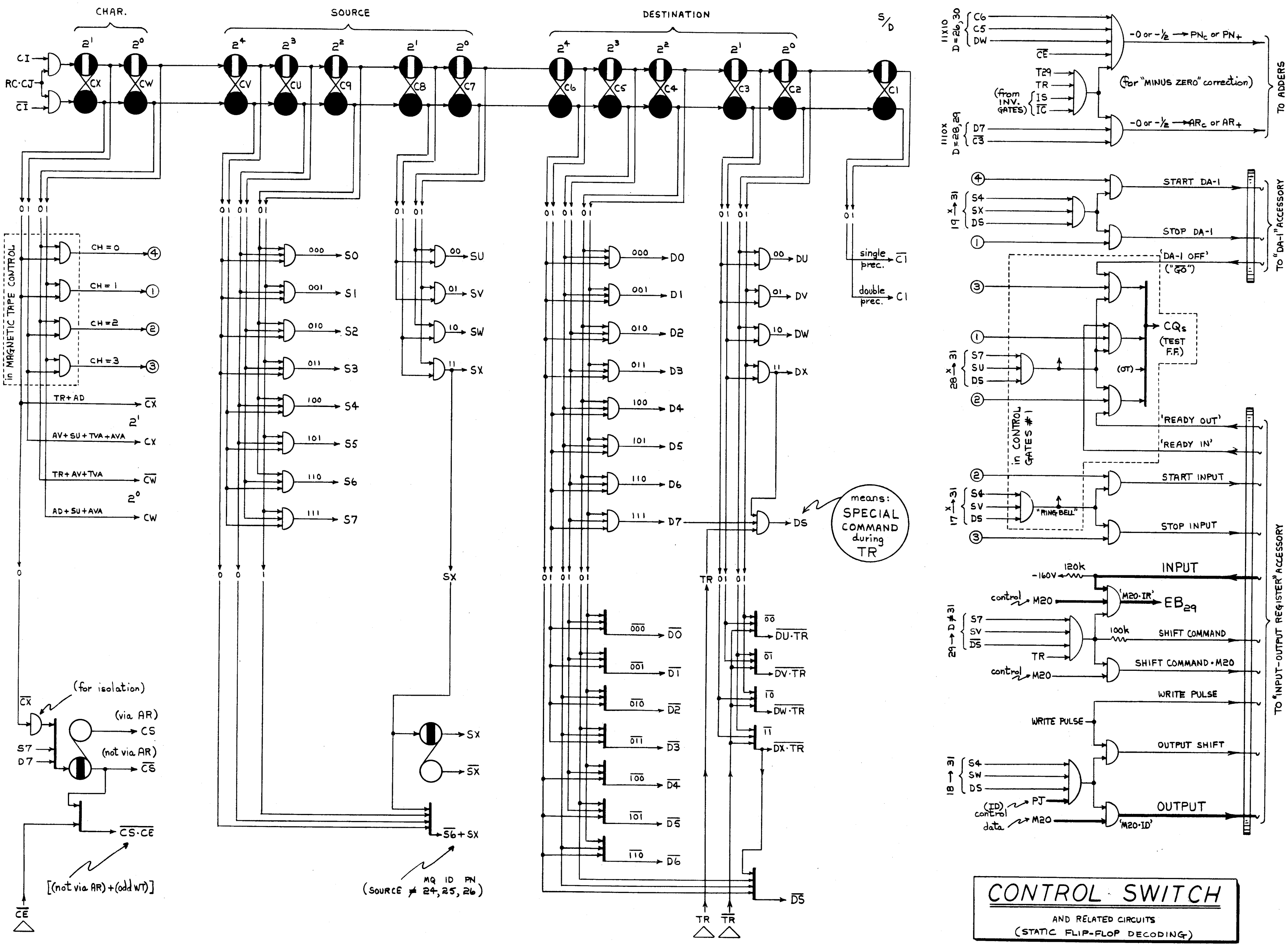
LINES 14, 16 & 17 RECIRCULATE AT ALL TIMES EXCEPT

1. At time of TRANSFER when the line is a DESTINATION
2. When the DA is in the GO condition

MEMORY LINES 0,1,2- - - -18
(RECIRCULATION, as SOURCES, as DESTINATIONS)



READ COMMAND



CONTROL SWITCH

AND RELATED CIRCUITS

(STATIC FLIP-FLOP DECODING)

ARITHMETIC OPERATIONS ON FRACTIONAL NUMBERS - GENERAL CASES

I. ABSOLUTE VALUE AND SIGN

AUGEND $0 \leqslant a < 1$	ADDEND $0 \leqslant b < 1$	SUM (ALL)	SUM (SPLIT)			
$+a$	$+b$	$0 \leqslant (+a+b) < 2$	$1 < (+a+b) < 2$	or	$+(a+b) > 1$	O.F.
			$(+a+b) = 1$	or	$+(a+b) = 1$	
			$0 \leqslant (+a+b) < 1$	or	$+(a+b) < 1$	
	$-b$	$-1 < (+a-b) < 1$	$0 \leqslant (+a-b) < 1$	or	$+(a-b) < 1$	
			$-1 < (+a-b) < 0$	or	$-(b-a) > -1$	
$-a$	$+b$	$-1 < (-a+b) < 1$	$0 \leqslant (-a+b) < 1$	or	$+(b-a) < 1$	
			$-1 < (-a+b) < 0$	or	$-(a-b) > -1$	
	$-b$	$-2 < (-a-b) < 0$	$-1 < (-a-b) < 0$	or	$-(a+b) > -1$	
			$(-a-b) = -1$	or	$-(a+b) = -1$	O.F.
			$-2 < (-a-b) < -1$	or	$-(a+b) < -1$	

II. ADDITION OF COMPLEMENTS

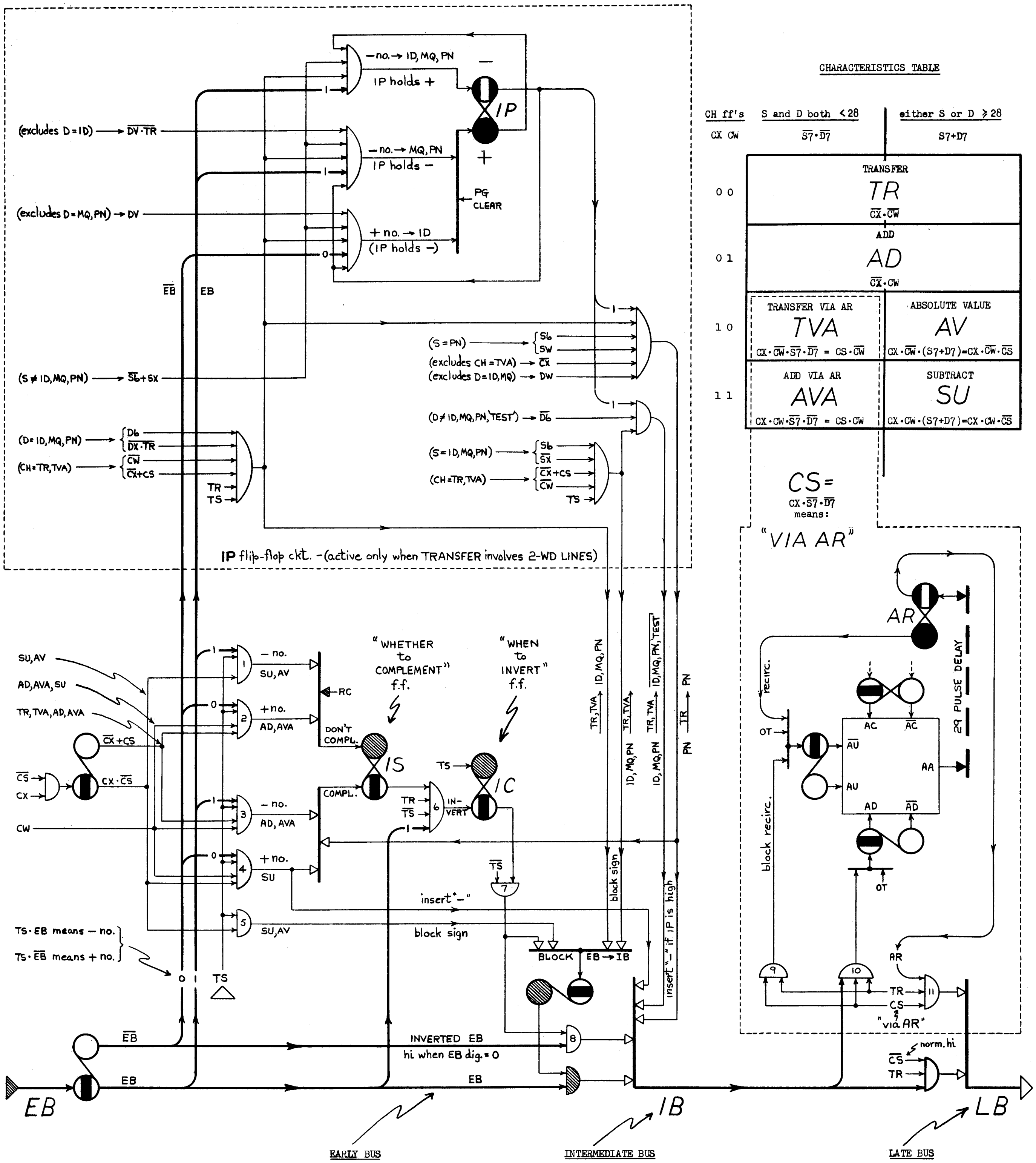
AUGEND $0 \leqslant a < 1$	ADDEND $0 \leqslant b < 1$	SUM (ALL)	SUM (SPLIT)	END CARRY	SUM LESS END CARRY		
$+a$	$+b$	$0 \leqslant (a+b) < 2$	$1 < (a+b) < 2$	C	$(a+b)-1$	0 0 0 1 1 (-)	O.F.
			$(a+b) = 1$	C	$(a+b)-1=0$	0 0 0 1 1 (-)	
			$0 \leqslant (a+b) < 1$		$(a+b)$	0 0 0 0 0 (+)	
	$+(1-b)$	$0 < (1+a-b) < 2$	$1 \leqslant (1+a-b) < 2$	C	$(a-b)$	0 1 1 1 0 (+)	
			$0 < (1+a-b) < 1$		$1-(b-a)$	0 1 1 0 1 (-)	
$+(1-a)$	$+b$	$0 < (1-a+b) < 2$	$1 \leqslant (1-a+b) < 2$	C	$(b-a)$	1 0 1 1 0 (+)	
			$0 < (1-a+b) < 1$		$1-(a-b)$	1 0 1 0 1 (-)	
	$+(1-b)$	$0 < (2-a-b) < 2$	$1 < (2-a-b) < 2$	C	$1-(a+b)$	1 1 0 1 1 (-)	
			$(2-a-b) = 1$	C	$1-(a+b)=0$	1 1 0 1 1 (-)	O.F.
			$0 < (2-a-b) < 1$		$2-(a+b)$	1 1 0 0 0 (+)	

sign of augend
sign of addend
uncorrected sign
end carry
corrected sign

```
DECIMAL   BINARY EQUIV.              FROM MEMORY    IN ADDER        TO MEMORY

                                           TS             TS
  +9        +1001                      1001/0  ───>  1001/0
  +3        +0011                      0011/0  ───>  0011/0
 ----      ------                                    ------
 +12        +1100 = +12                              1100/0
                                                       /0c
                                                     ------
                                                     1100/0  ───>  1100/0 = +12

  +9        +1001                      1001/0  ───>  1001/0
  -3        -0011                      0011/1  ───>  1101/1
 ----      ------                                    ------
  +6        +0110 = +6                               0110/1
                                                       /1c
                                                     ------
                                                     0110/0  ───>  0110/0 = +6

  -9        -1001                      1001/1  ───>  0111/1
  +3        +0011                      0011/0  ───>  0011/0
 ----      ------                                    ------
  -6        -0110 = -6                               1010/1
                                                       /0c
                                                     ------
                                                     1010/1  ───>  0110/1 = -6

  -9        -1001                      1001/1  ───>  0111/1
  -3        -0011                      0011/1  ───>  1101/1
 ----      ------                                    ------
 -12        -1100 = -12                              0100/0
                                                       /1c
                                                     ------
                                                     0100/1  ───>  1100/1 = -12
```

OVERLOAD EXAMPLES

```
  +8        +1000                      1000/0  ───>  1000/0
  +9        +1001                      1001/0  ───>  1001/0        "sum ≥ 1"
 ----      ------                                    ------
 +17       +10001 (= +17)                            0001/0
            ≡                                          /1c
                                                     ------
                                                     0001/1  ───>  1111/1 = -15  X

  -8        -1000                      1000/1  ───>  1000/1
  -9        -1001                      1001/1  ───>  0111/1        "|-sum| > 1"
 ----      ------                                    ------
 -17       -10001 (= -17)                            1111/0
            ≡                                          /0c
                                                     ------
                                                     1111/0  ───>  1111/0 = +15  X

  -7        -0111                      0111/1  ───>  1001/1
  -9        -1001                      1001/1  ───>  0111/1        "|-sum| = 1"
 ----      ------                                    ------
 -16       -10000 (= -16)                            0000/0
            ≡                                          /1c
                                                     ------
                                                     0000/1  ───>  0000/1 = -0  X
```



CHARACTERISTICS TABLE

CH ff's CX CW	S and D both < 28 $\overline{S7} \cdot \overline{D7}$	either S or D ⩾ 28 S7+D7
0 0	TRANSFER TR $\overline{CX} \cdot \overline{CW}$	
0 1	ADD AD $\overline{CX} \cdot CW$	
1 0	TRANSFER VIA AR TVA $CX \cdot \overline{CW} \cdot \overline{S7} \cdot \overline{D7} = CS \cdot \overline{CW}$	ABSOLUTE VALUE AV $CX \cdot \overline{CW} \cdot (S7+D7) = CX \cdot \overline{CW} \cdot \overline{CS}$
1 1	ADD VIA AR AVA $CX \cdot CW \cdot \overline{S7} \cdot \overline{D7} = CS \cdot CW$	SUBTRACT SU $CX \cdot CW \cdot (S7+D7) = CX \cdot CW \cdot \overline{CS}$

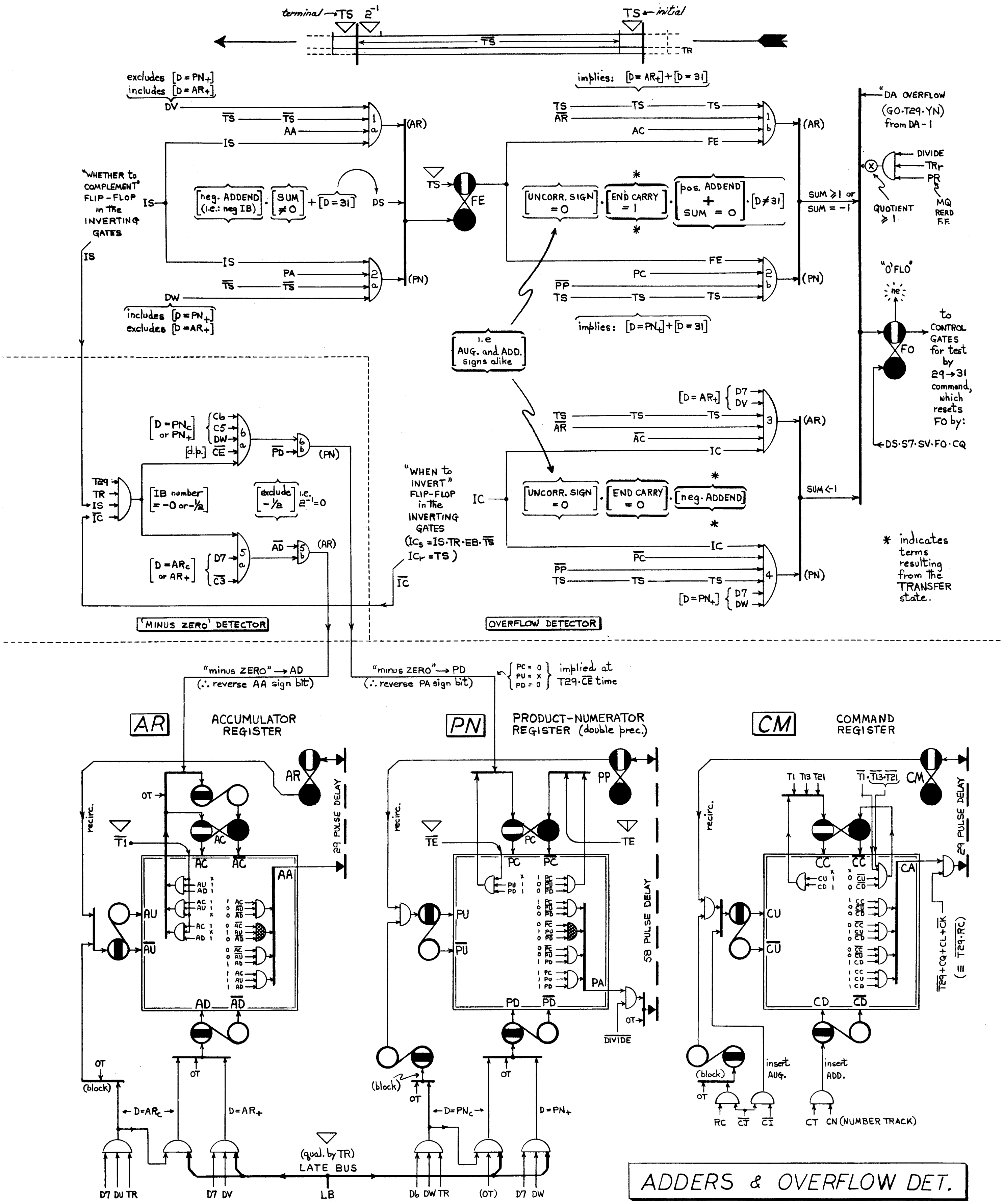
CS = $CX \cdot \overline{S7} \cdot \overline{D7}$ means: "VIA AR"

		EARLY BUS		INTERMEDIATE BUS	
CHARACTERISTIC	EB→IB PROCESS	DECIMAL	BINARY	BINARY	DECIMAL
TRANSFER or TRANSFER VIA AR	PASS UNCHANGED (if 2-WORD LINES are not involved)	$+ 12 \times 2^{-8}$ = $- 12 \times 2^{-8}$ =	.0 0 0 0 1 1 0 0/0 .0 0 0 0 1 1 0 0/1	.0 0 0 0 1 1 0 0/0 = .0 0 0 0 1 1 0 0/1 =	$+ 12 \times 2^{-8}$ $- 12 \times 2^{-8}$
ADD or ADD VIA AR	COMPLEMENT NEGATIVE NUMBERS	$+ 12 \times 2^{-8}$ = $- 12 \times 2^{-8}$ =	.0 0 0 0 1 1 0 0/0 .0 0 0 0 1 1 0 0/1	.0 0 0 0 1 1 0 0/0 = .1 1 1 1 0 1 0 0/1 =	$+ 12 \times 2^{-8}$ $1 - 12 \times 2^{-8}$
ABSOLUTE VALUE	BLOCK SIGN	$+ 12 \times 2^{-8}$ = $- 12 \times 2^{-8}$ =	.0 0 0 0 1 1 0 0/0 .0 0 0 0 1 1 0 0/1	.0 0 0 0 1 1 0 0/● = .0 0 0 0 1 1 0 0/● =	12×2^{-8} 12×2^{-8}
SUBTRACT	REVERSE SIGN and COMPLEMENT NUMBERS if NEW SIGN is NEG.	$+ 12 \times 2^{-8}$ = $- 12 \times 2^{-8}$ =	.0 0 0 0 1 1 0 0/0 .0 0 0 0 1 1 0 0/1	.1 1 1 1 0 1 0 0/1 = .0 0 0 0 1 1 0 0/0 =	$1 - 12 \times 2^{-8}$ $+ 12 \times 2^{-8}$

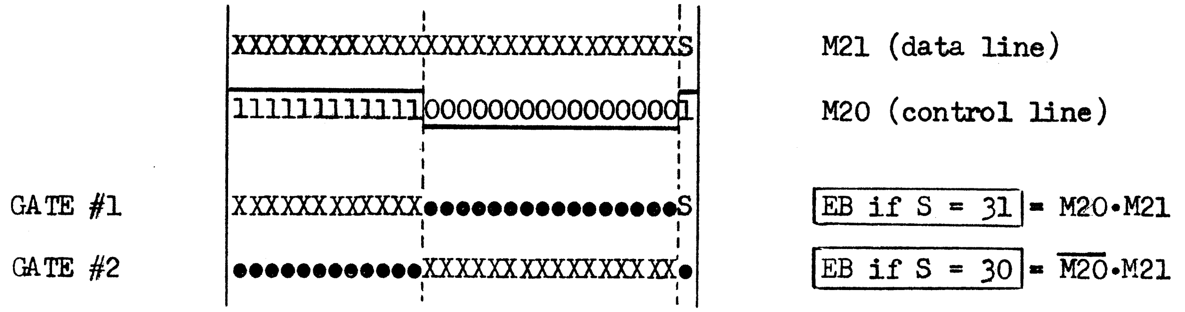
EB ⟶ IB ⟶ LB

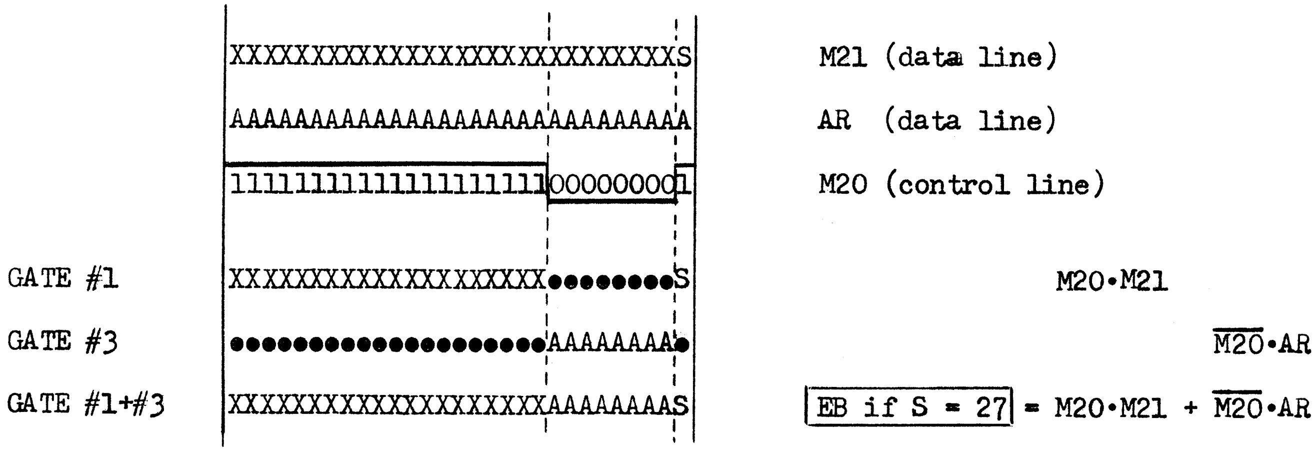
INVERTING GATES (complete) | ACC. GATES (part)

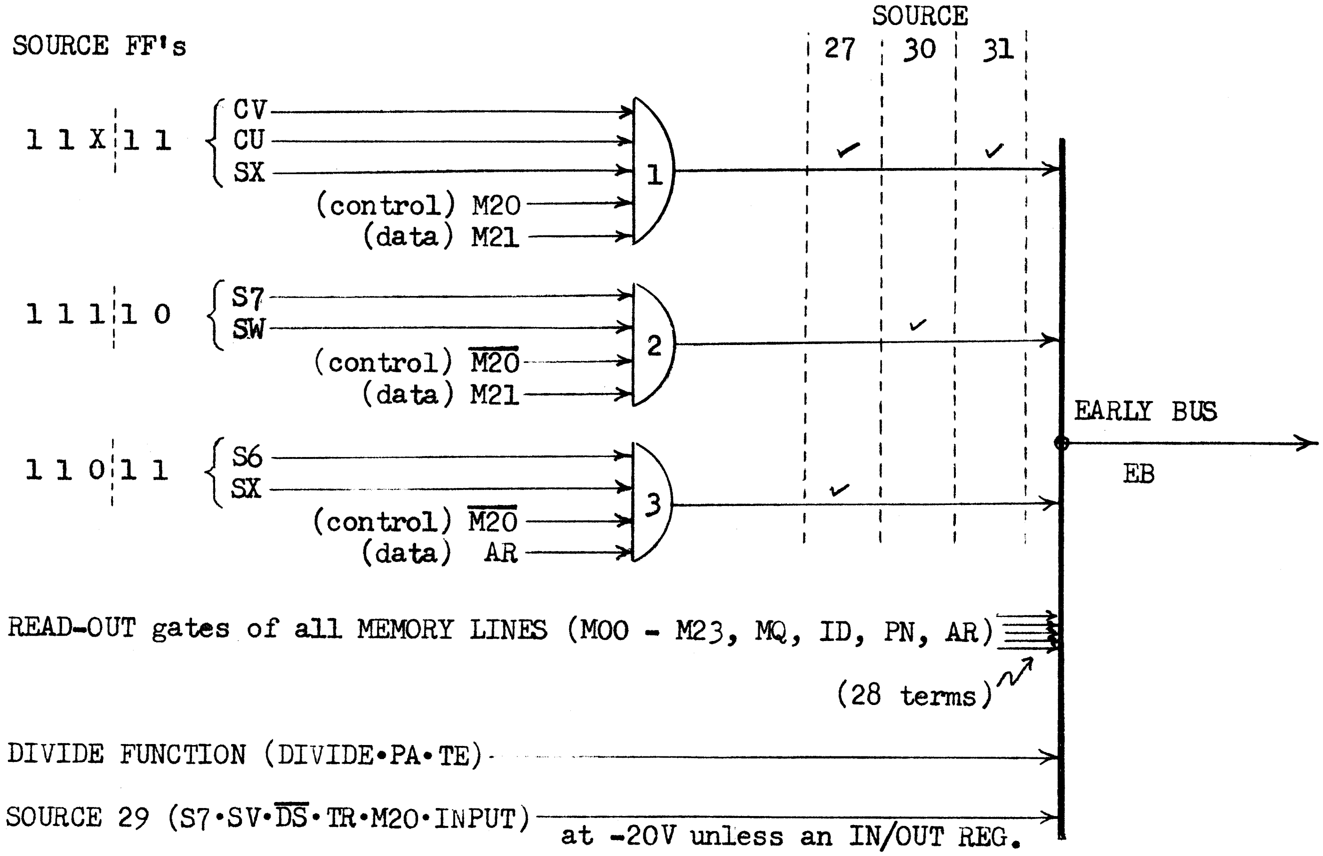
G-15D



ADDERS & OVERFLOW DET.

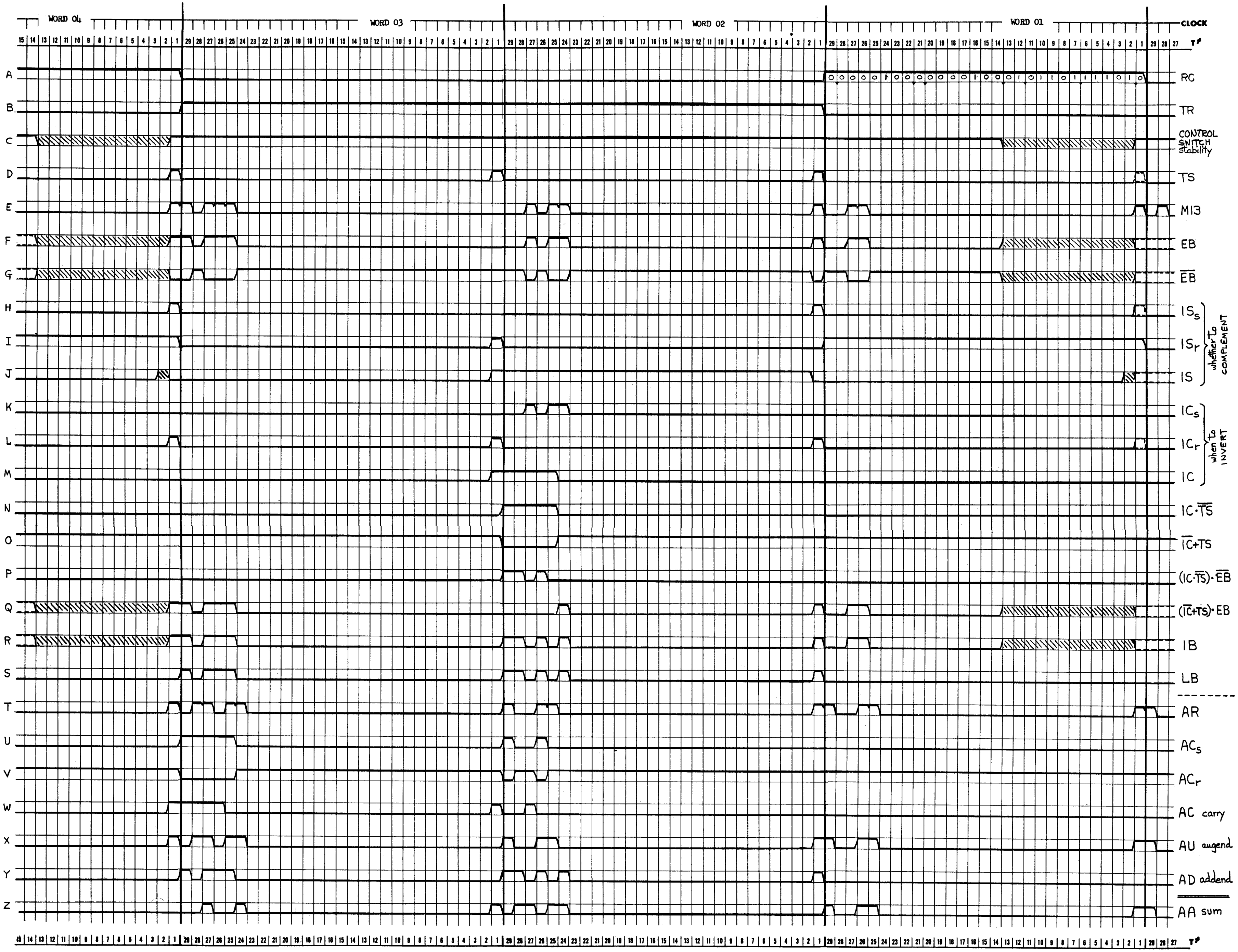




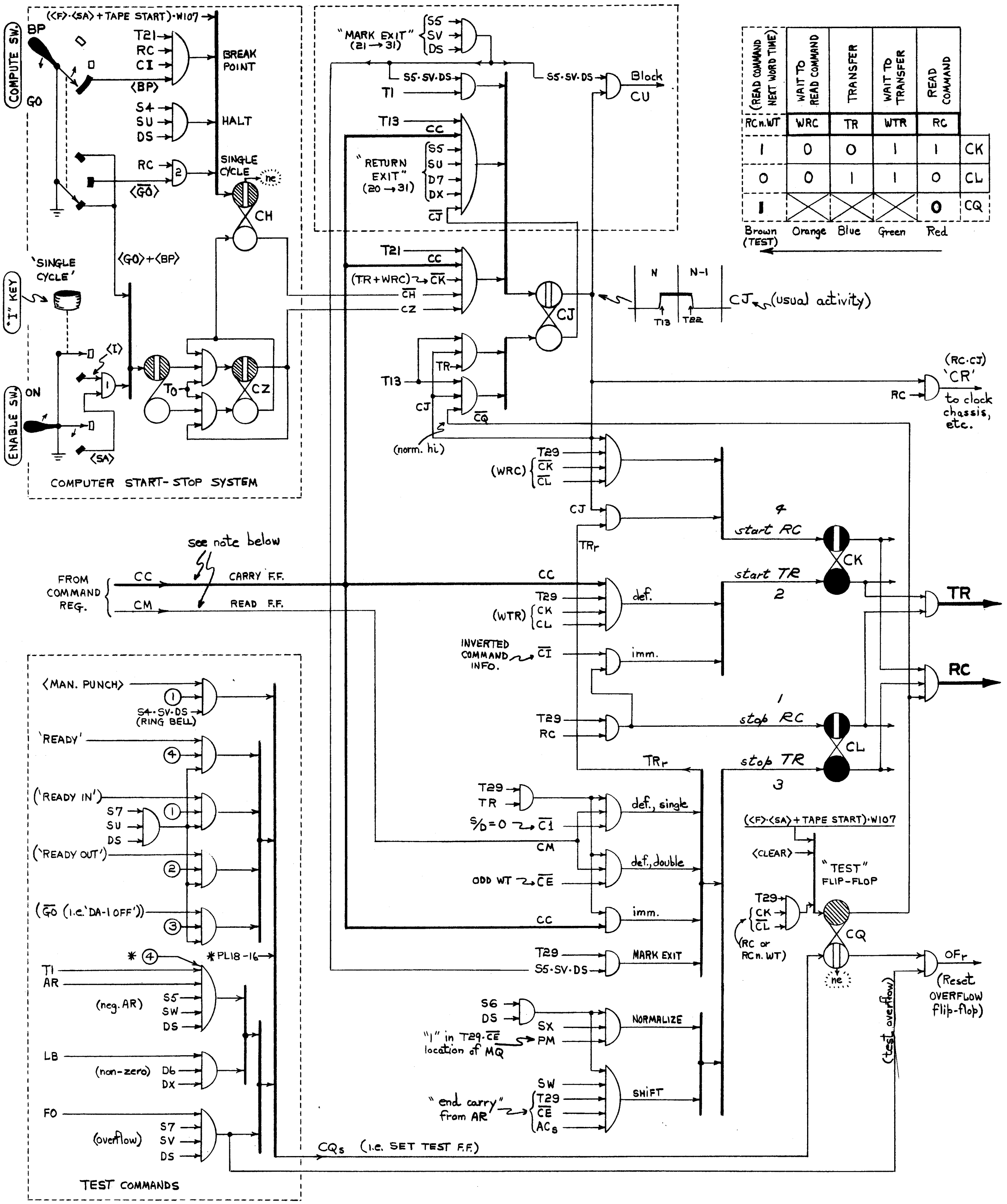

SOURCE FF's
SOURCE
27
30
31
1 1 X 1 1
CV
CU
SX
(control) M20
(data) M21
1
1 1 1 1 0
S7
SW
(control) $\overline{\text{M20}}$
(data) M21
2
1 1 0 1 1
S6
SX
(control) $\overline{\text{M20}}$
(data) AR
3
EARLY BUS
EB
READ-OUT gates of all MEMORY LINES (M00 - M23, MQ, ID, PN, AR)
(28 terms)
DIVIDE FUNCTION (DIVIDE•PA•TE)
SOURCE 29 (S7•SV•$\overline{\text{DS}}$•TR•M20•INPUT)

at -20V unless an IN/OUT REG. accessory is plugged in. This is a useful source of ZEROS.

COMPLEX SOURCES & EARLY BUS



WAVEFORMS: EB, IB, LB, AR, etc.
example: (13.02-03) $\xrightarrow{AD} AR_+$



(READ COMMAND NEXT WORD TIME)	WAIT TO READ COMMAND	TRANSFER	WAIT TO TRANSFER	READ COMMAND	
RCn.WT	WRC	TR	WTR	RC	
1	0	0	1	1	CK
0	0	1	1	0	CL
1	☒	☒	☒	0	CQ
Brown (TEST)	Orange	Blue	Green	Red	

NOTE:

COMMAND REG. OUTPUTS

CC

- CC · T29 · $\overline{RC}$ (Abs. T#) occurs at WT =(T-1)
- CC · T29 · $\overline{RC}$ (Rel. T#) " " WT =(L+T)
- CC · T21 · $\overline{RC}$ ———— " " WT =(N-1)
- CC · T13 ———— " " WT =(TR-1) of previous 'MARK EXIT' command

CM : CM · T29 follows first CC · T29 · $\overline{RC}$ for 108 WT's

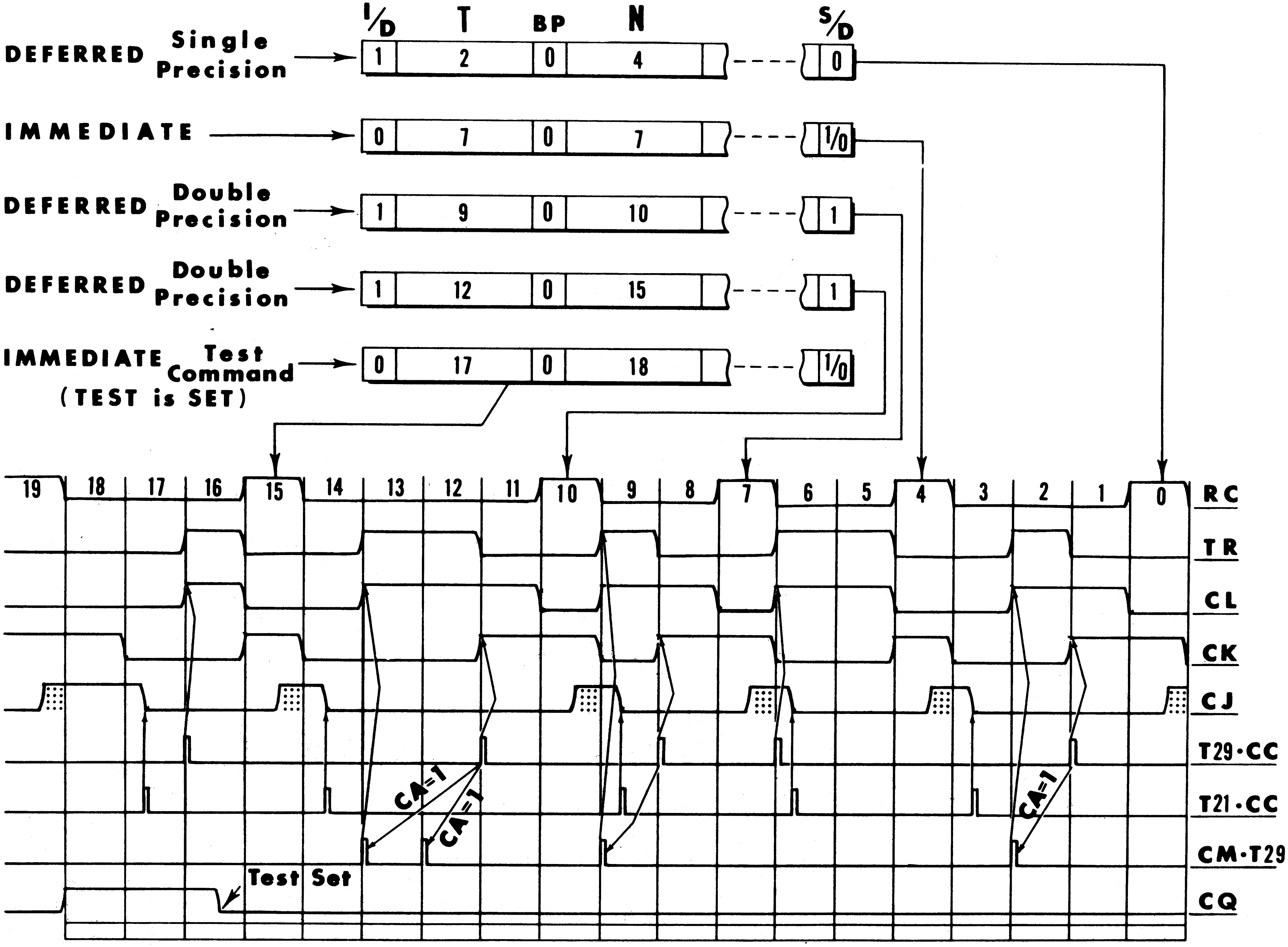
* Applied to all models with ser. #'s ≥ 211 and to earlier models modified for CA-2

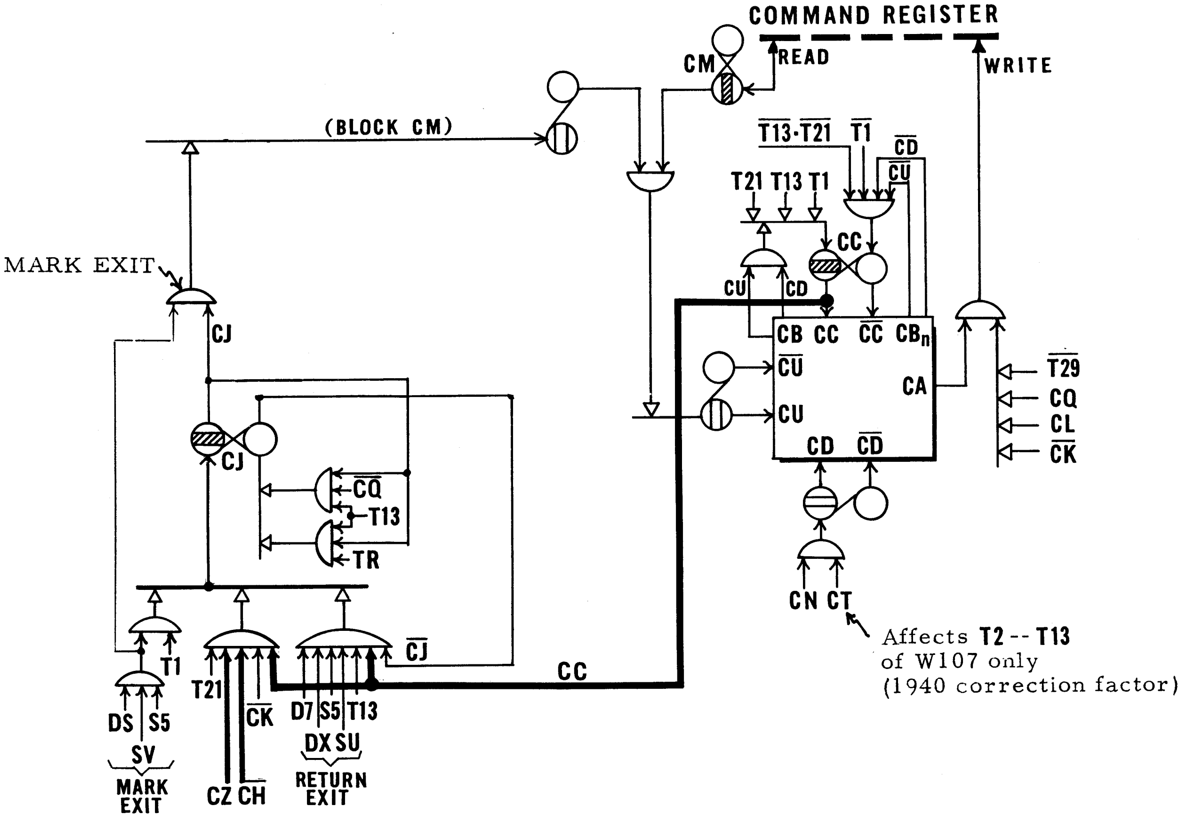
CONTROL GATES

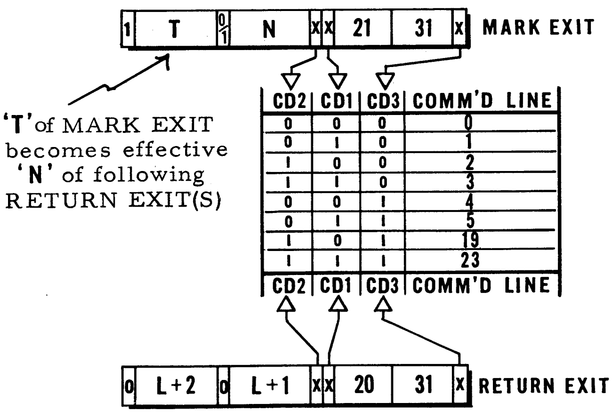
(STATES OF COMPUTER)

G15D

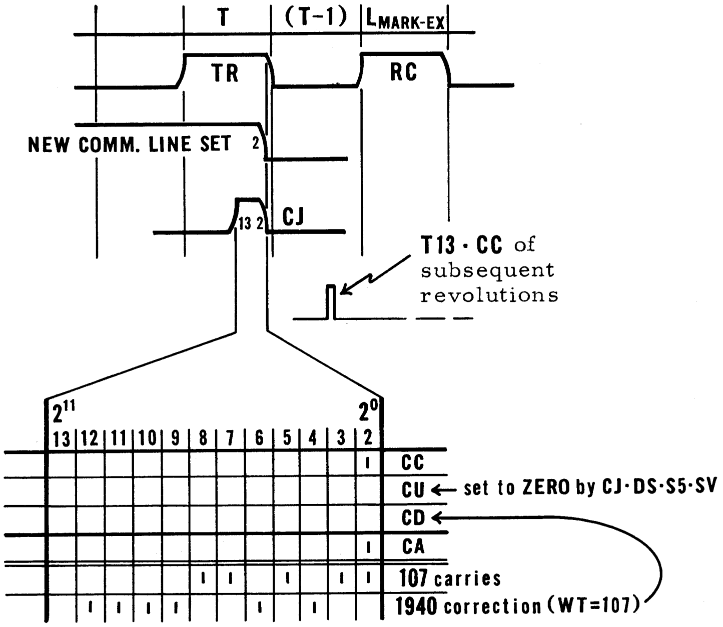
COMMAND READ





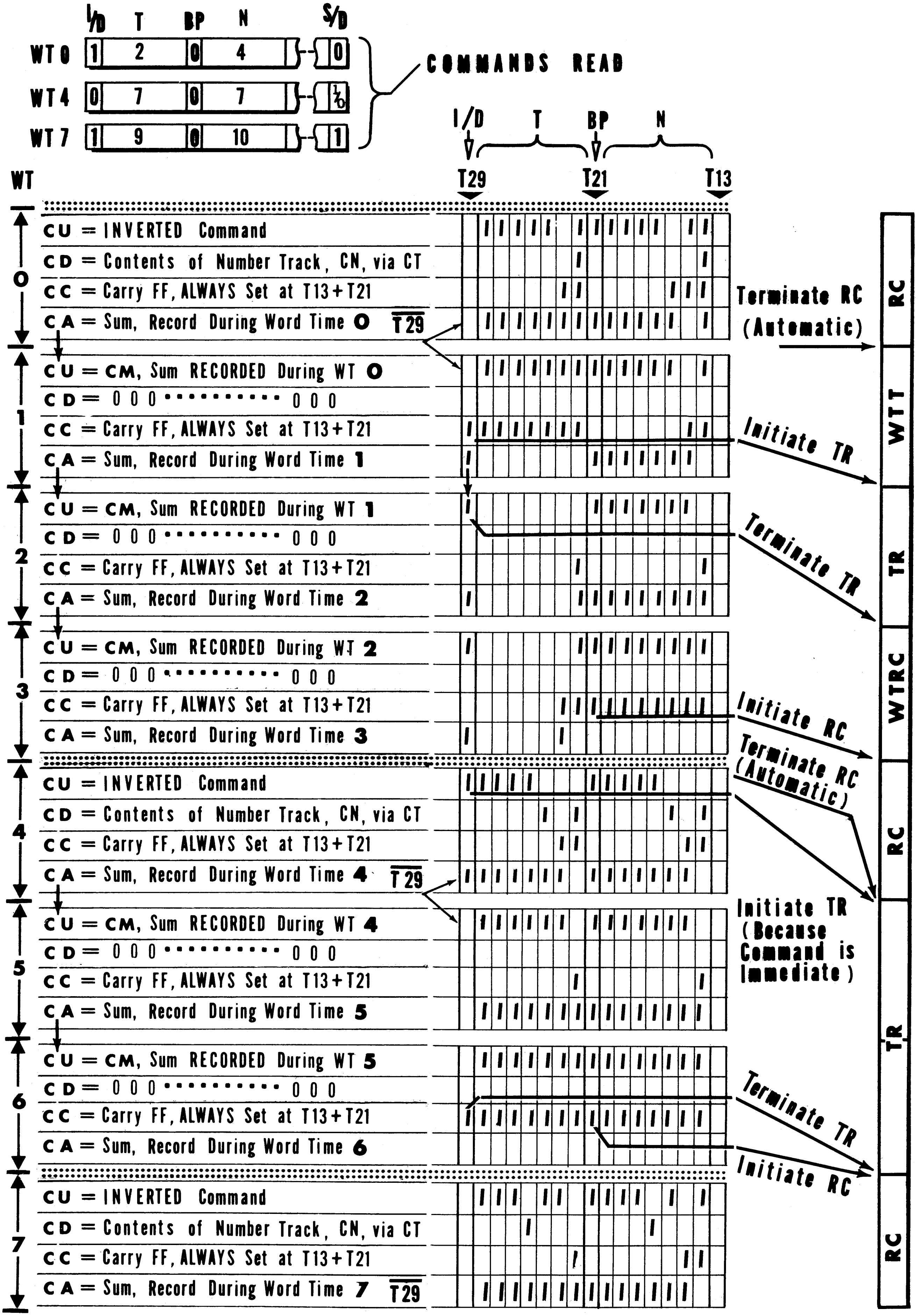

1 T 0/1 N x x 21 31 x MARK EXIT
'T' of MARK EXIT becomes effective 'N' of following RETURN EXIT(S)
0 L+2 0 L+1 x x 20 31 x RETURN EXIT

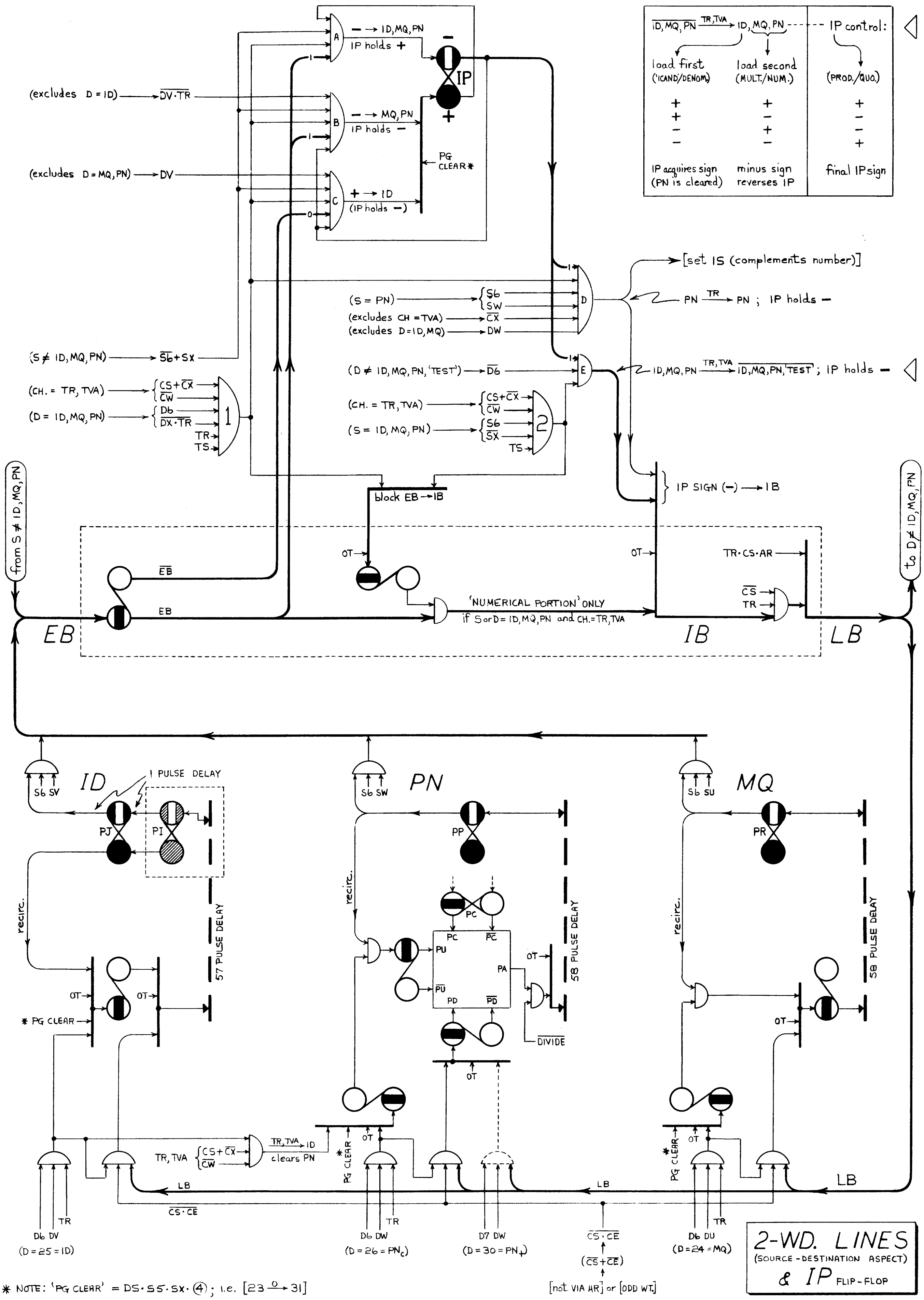
CD2	CD1	CD3	COMM'D LINE
0	0	0	0
0	1	0	1
1	0	0	2
1	1	0	3
0	0	1	4
0	1	1	5
1	0	1	19
1	1	1	23
CD2	CD1	CD3	COMM'D LINE



	13	12	11	10	9	8	7	6	5	4	3	2
CC												1
CU												
CD												
CA												1
107 carries						1	1		1		1	1
1940 correction		1	1	1	1			1		1		

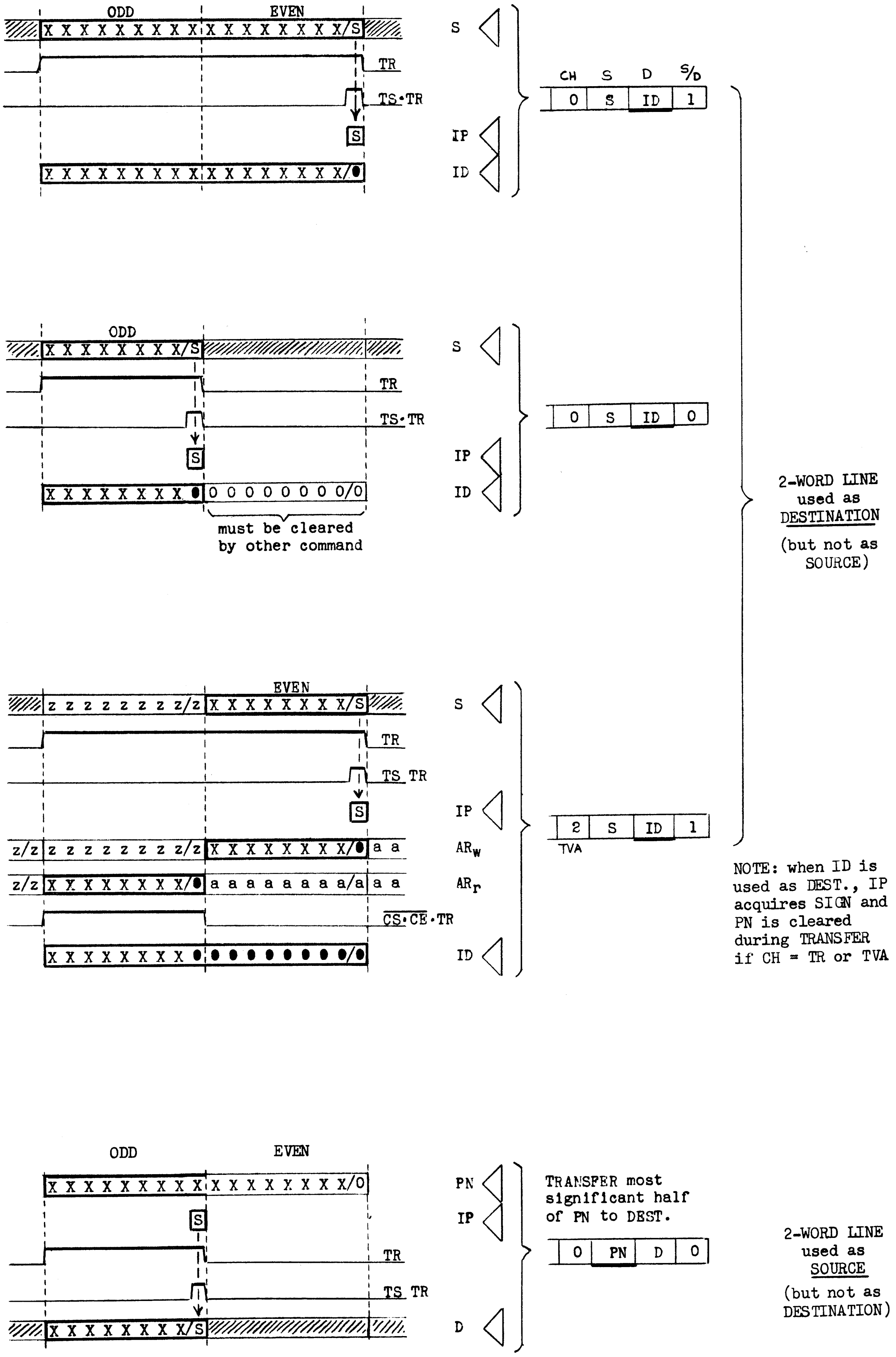
MARK & RETURN EXIT


I/D T BP N S/D
WT 0 1 2 0 4 0
WT 4 0 7 0 7 1/0
WT 7 1 9 0 10 1
COMMANDS READ
I/D T BP N
T29 T21 T13
WT
0
CU = INVERTED Command
CD = Contents of Number Track, CN, via CT
CC = Carry FF, ALWAYS Set at T13+T21
CA = Sum, Record During Word Time 0 T29
1
CU = CM, Sum RECORDED During WT 0
CD = 000 ··········· 000
CC = Carry FF, ALWAYS Set at T13+T21
CA = Sum, Record During Word Time 1
2
CU = CM, Sum RECORDED During WT 1
CD = 000 ··········· 000
CC = Carry FF, ALWAYS Set at T13+T21
CA = Sum, Record During Word Time 2
3
CU = CM, Sum RECORDED During WT 2
CD = 000 ··········· 000
CC = Carry FF, ALWAYS Set at T13+T21
CA = Sum, Record During Word Time 3
4
CU = INVERTED Command
CD = Contents of Number Track, CN, via CT
CC = Carry FF, ALWAYS Set at T13+T21
CA = Sum, Record During Word Time 4 T29
5
CU = CM, Sum RECORDED During WT 4
CD = 000 ··········· 000
CC = Carry FF, ALWAYS Set at T13+T21
CA = Sum, Record During Word Time 5
6
CU = CM, Sum RECORDED During WT 5
CD = 000 ··········· 000
CC = Carry FF, ALWAYS Set at T13+T21
CA = Sum, Record During Word Time 6
7
CU = INVERTED Command
CD = Contents of Number Track, CN, via CT
CC = Carry FF, ALWAYS Set at T13+T21
CA = Sum, Record During Word Time 7 T29
Terminate RC (Automatic)
Initiate TR
Terminate TR
Initiate RC
Terminate RC (Automatic)
Initiate TR (Because Command is Immediate)
Terminate TR
Initiate RC
RC
WTT
TR
WTRC
RC
TR
RC



2-WD. LINES
(SOURCE-DESTINATION ASPECT)
& IP FLIP-FLOP

* NOTE: 'PG CLEAR' = DS·S5·SX·④ ; i.e. [23 $\xrightarrow{0}$ 31]


ODD
EVEN
X X X X X X X X X X X X X X X X X X/S
TR
TS·TR
S
X X X X X X X X X X X X X X X X X X/●
S
IP
ID
CH S D S/D
0 S ID 1
ODD
X X X X X X X X/S
TR
TS·TR
S
X X X X X X X X ● 0 0 0 0 0 0 0 0 0/0
S
IP
ID
0 S ID 0
EVEN
z z z z z z z z/z X X X X X X X X X/S
TR
TS TR
S
z/z z z z z z z z z/z X X X X X X X X/● a a
z/z X X X X X X X X/● a a a a a a a a/a a a
CS·CE·TR
X X X X X X X X ● ● ● ● ● ● ● ● ● ●/●
S
IP
ARw
ARr
ID
2 S ID 1
TVA
ODD
EVEN
X X X X X X X X X X X X X X X X X X/0
S
TR
TS TR
X X X X X X X X/S
PN
IP
D
0 PN D 0

must be cleared by other command

2-WORD LINE used as DESTINATION

(but not as SOURCE)

NOTE: when ID is used as DEST., IP acquires SIGN and PN is cleared during TRANSFER if CH = TR or TVA

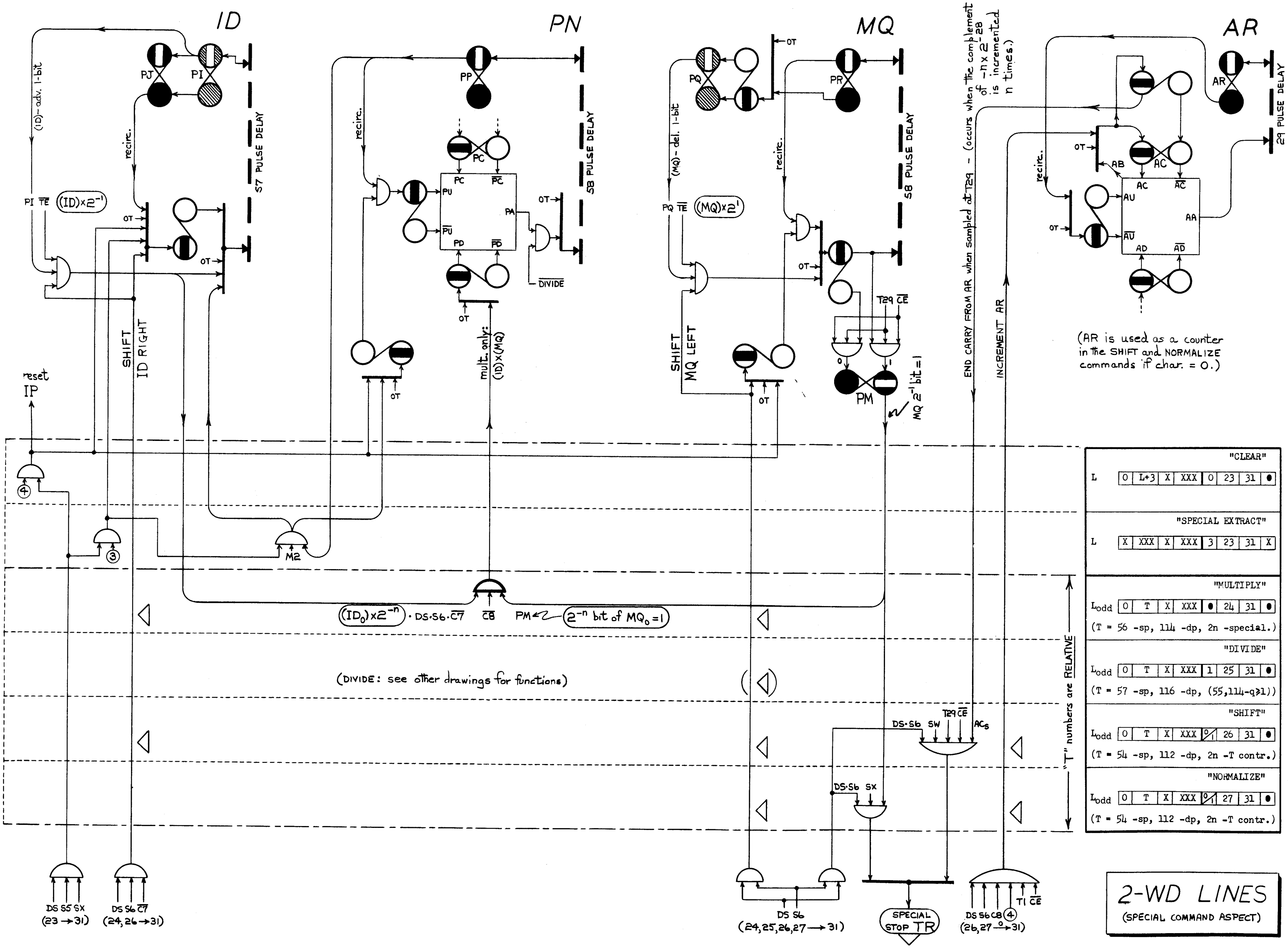
TRANSFER most significant half of PN to DEST.

2-WORD LINE used as SOURCE

(but not as DESTINATION)

2-WORD LINES

SOURCE, DESTINATION examples



"CLEAR"

L	0	L+3	X	XXX	0	23	31	●

"SPECIAL EXTRACT"

L	X	XXX	X	XXX	3	23	31	X

"MULTIPLY"

L_{odd}	0	T	X	XXX	●	24	31	●

(T = 56 -sp, 114 -dp, 2n -special.)

"DIVIDE"

L_{odd}	0	T	X	XXX	1	25	31	●

(T = 57 -sp, 116 -dp, (55,114-q≥1))

"SHIFT"

L_{odd}	0	T	X	XXX	0/1	26	31	●

(T = 54 -sp, 112 -dp, 2n -T contr.)

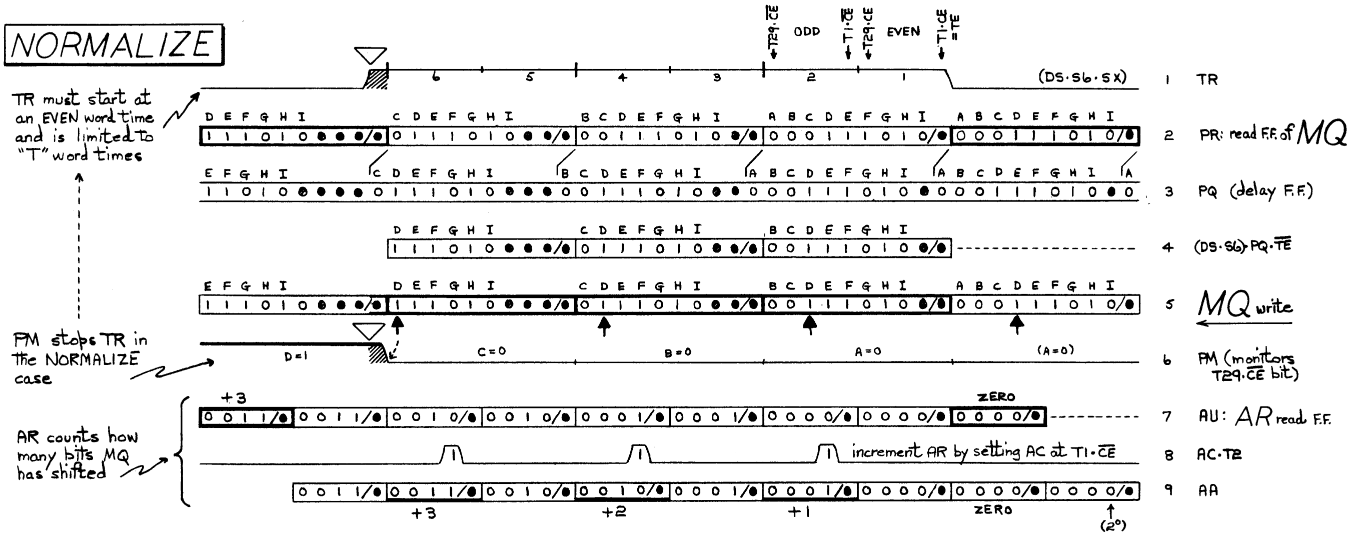
"NORMALIZE"

L_{odd}	0	T	X	XXX	0/1	27	31	●

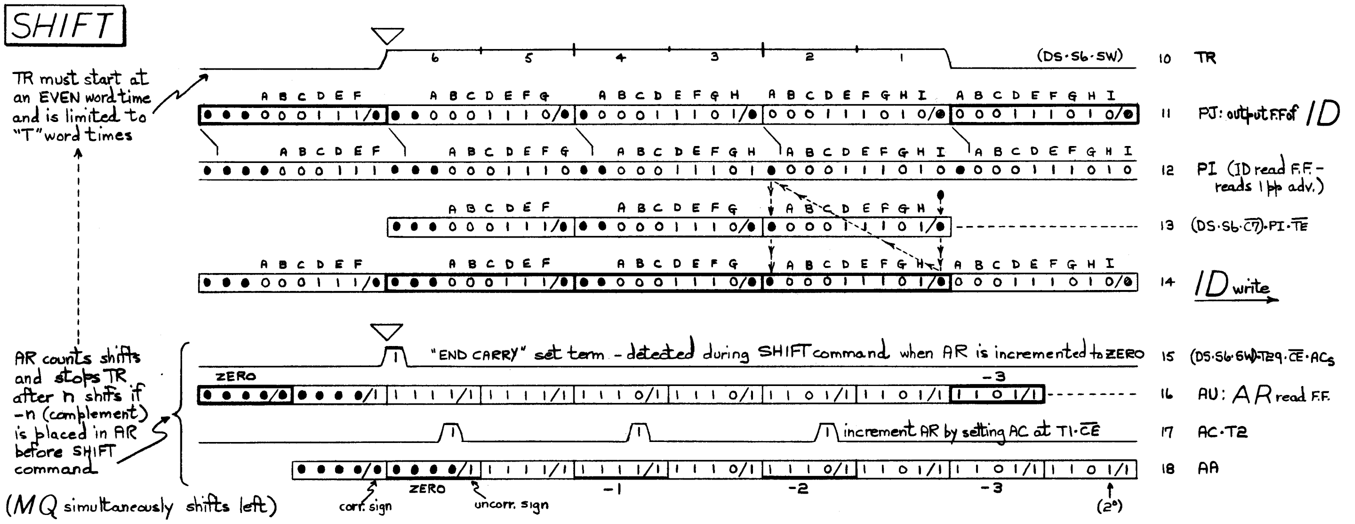
(T = 54 -sp, 112 -dp, 2n -T contr.)

2-WD LINES
(SPECIAL COMMAND ASPECT)

NORMALIZE



SHIFT



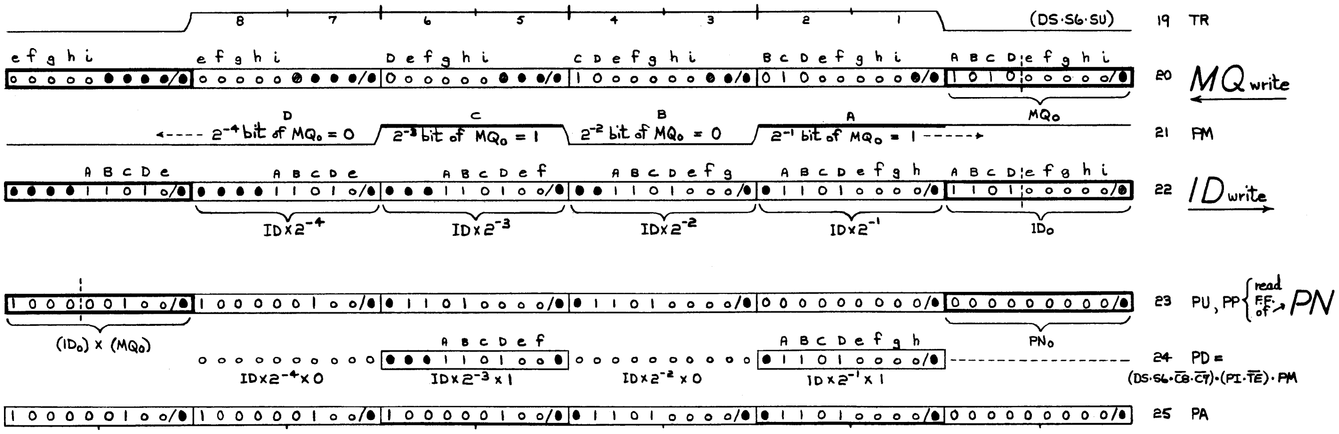
6 5 4 3 2 1
(DS·S6·SW) 10 TR
TR must start at an EVEN word time and is limited to "T" word times
A B C D E F | A B C D E F G | A B C D E F G H | A B C D E F G H I | A B C D E F G H I
● ● ● ● 0 0 0 1 1 1/● | ● ● ● 0 0 0 1 1 1 0/● | ● ● 0 0 0 1 1 1 0 1/● | 0 0 0 1 1 1 0 1 0/● | 0 0 0 1 1 1 0 1 0/●
11 PJ: output F.F. of ID
12 PI (ID read F.F. - reads 1 pp adv.)
13 (DS·S6·C̄7̄)·PI·T̄E
14 ID write
"END CARRY" set term - detected during SHIFT command when AR is incremented to ZERO
15 (DS·S6·SW)·T29·C̄Ē·AC5
AR counts shifts and stops TR after n shifts if -n (complement) is placed in AR before SHIFT command
ZERO -3
● ● ● ●/● | ● ● ● ●/1 | 1 1 1 1/1 | 1 1 1 1/1 | 1 1 1 0/1 | 1 1 1 0/1 | 1 1 0 1/1 | 1 1 0 1/1 | 1 1 0 1/1
16 AU: AR read F.F.
increment AR by setting AC at T1·C̄Ē
17 AC·T2
● ● ● ●/● | ● ● ● ●/1 | 1 1 1 1/1 | 1 1 1 1/1 | 1 1 1 0/1 | 1 1 1 0/1 | 1 1 0 1/1 | 1 1 0 1/1 | 1 1 0 1/1
corr. sign ZERO uncorr. sign -1 -2 -3 (2⁰)
18 AA
(MQ simultaneously shifts left)

MULTIPLY

$\frac{13}{16} \times \frac{10}{16} = \frac{130}{256}$ $\frac{8}{16}$

example given- .1101 × .1010 = .10000010 (round-off to .1000)

(ID) × (MQ) ⟶ PN



8 7 6 5 4 3 2 1
(DS·S6·SU) 19 TR
e f g h i | e f g h i | D e f g h i | C D e f g h i | B C D e f g h i | A B C D e f g h i
20 MQ write
MQ₀
D C B A
2⁻⁴ bit of MQ₀ = 0 | 2⁻³ bit of MQ₀ = 1 | 2⁻² bit of MQ₀ = 0 | 2⁻¹ bit of MQ₀ = 1
21 PM
A B C D e | A B C D e | A B C D e f | A B C D e f g | A B C D e f g h | A B C D e f g h i
22 ID write
ID×2⁻⁴ ID×2⁻³ ID×2⁻² ID×2⁻¹ ID₀
1 0 0 0 0 0 1 0 0/● | 1 0 0 0 0 0 1 0 0/● | ● 1 1 0 1 0 0 0 0/● | ● 1 1 0 1 0 0 0 0/● | 0 0 0 0 0 0 0 0 0/● | 0 0 0 0 0 0 0 0 0/●
23 PU, PP {read F.F. of} PN
(ID₀) × (MQ₀) PN₀
A B C D e f | A B C D e f g h
ID×2⁻⁴×0 ID×2⁻³×1 ID×2⁻²×0 ID×2⁻¹×1
24 PD = (DS·S6·C̄8̄·C̄7̄)·(PI·T̄E)·PM
1 0 0 0 0 0 1 0 0/● | 1 0 0 0 0 0 1 0 0/● | 1 0 0 0 0 0 1 0 0/● | ● 1 1 0 1 0 0 0 0/● | ● 1 1 0 1 0 0 0 0/● | 0 0 0 0 0 0 0 0 0/●
25 PA

NUMERICAL EXAMPLES

NORMALIZE, SHIFT, & MULTIPLY

DIVISION ($q = \frac{n}{d} < 2$; i.e. $n < 2d$)

			q			if
	d		1.0100			$d_n > r_n$
(A)	.10100● /		.11001●●●●●●●	n	$0 \leq (n = r_0) < 1$	
	1	sub	.10100●●●●●●●	$d(2^0)$ x 1	$\frac{1}{2} \leq (d = d_0) < 1$	or 0
	•	+	.00101●●●●●●●	$r_1 = n - d(2^0)$	$0 \leq r_1 < d_0 < 1$	
		+	.001010●●●●●●	"	"	
	0	(sub)	.●00000●●●●●●	$d(2^{-1})$ x 0	$\frac{1}{4} \leq d_1 < \frac{1}{2}$	or 0
		+	.●01010●●●●●●	$r_2 = n - d(2^0)$	$0 \leq r_2 < d_1 < \frac{1}{2}$	
1.0100		+	.●010100●●●●●	"	"	
.10100/.11001	1	sub	.●●10100●●●●●	$d(2^{-2})$ x 1	$1/8 \leq d_2 < \frac{1}{4}$	or 0
-10100		+	.●●00000●●●●●	$r_3 = n - d(2^0+2^{-2})$	$0 \leq r_3 < d_2 < \frac{1}{4}$	
10100		+	.●●000000●●●●	"	"	
--10100	0	(sub)	.●●●00000●●●●	$d(2^{-3})$ x 0	$1/16 \leq d_3 < 1/8$	or 0
-0-		+	.●●●00000●●●●	$r_4 = n - d(2^0+2^{-2})$	$0 \leq r_4 < d_3 < 1/8$	
		+	.●●●000000●●●	"	"	
	0	(sub)	.●●●●00000●●●	$d(2^{-4})$ x 0	$1/32 \leq d_4 < 1/16$	or 0
		+	.●●●●00000●●●	$r_5 = n - d(2^0+2^{-2})$	$0 \leq r_5 < d_4 < 1/16$	
		+	.●●●●000000●●	"	"	

			q			
	d		1.0100			
(B)	.10100● /		.11001●●●●●●●	n	$0 \leq (n = r_0) < 1$	
		sub	.10100●●●●●●●	$d(2^0)$	$\frac{1}{2} \leq (d = d_0) < 1$	
1.0100	1	+	.00101●●●●●●●	$r_1 = n - d(2^0)$	$-1 < r_1 < \frac{1}{2}$	*
.10100/.11001	•	+	.001010●●●●●●	"	"	
- 10100		sub	.●10100●●●●●●	$d(2^{-1})$	$\frac{1}{4} \leq d_1 < \frac{1}{2}$	
1 +001010	0	-	.●01010●●●●●●	$r_2 = n - d(2^0+2^{-1})$	$-\frac{1}{2} < r_2 < \frac{1}{4}$	*
- 10100		-	.●010100●●●●●	"	"	
0 -010100		add	.●●10100●●●●●	$d(2^{-2})$	$1/8 \leq d_2 < \frac{1}{4}$	
+ 10100	1	+	.●●00000●●●●●	$r_3 = n - d(2^0+2^{-2})$	$-\frac{1}{4} < r_3 < 1/8$	*
1 +000000		+	.●●000000●●●●	"	"	
- 10100		sub	.●●●10100●●●●	$d(2^{-3})$	$1/16 \leq d_3 < 1/8$	
0 -101000	0	-	.●●●10100●●●●	$r_4 = n - d(2^0+2^{-2}+2^{-3})$	$-1/8 < r_4 < 1/16$	*
+ 10100		-	.●●●101000●●●	"	"	
0 -101000		add	.●●●●10100●●●	$d(2^{-4})$	$1/32 \leq d_4 < 1/16$	
	0	-	.●●●●10100●●●	$r_5 = n - d(2^0+2^{-2}+2^{-4})$	$-1/16 < r_5 < 1/32$	*
		-	.●●●●101000●●	"	"	

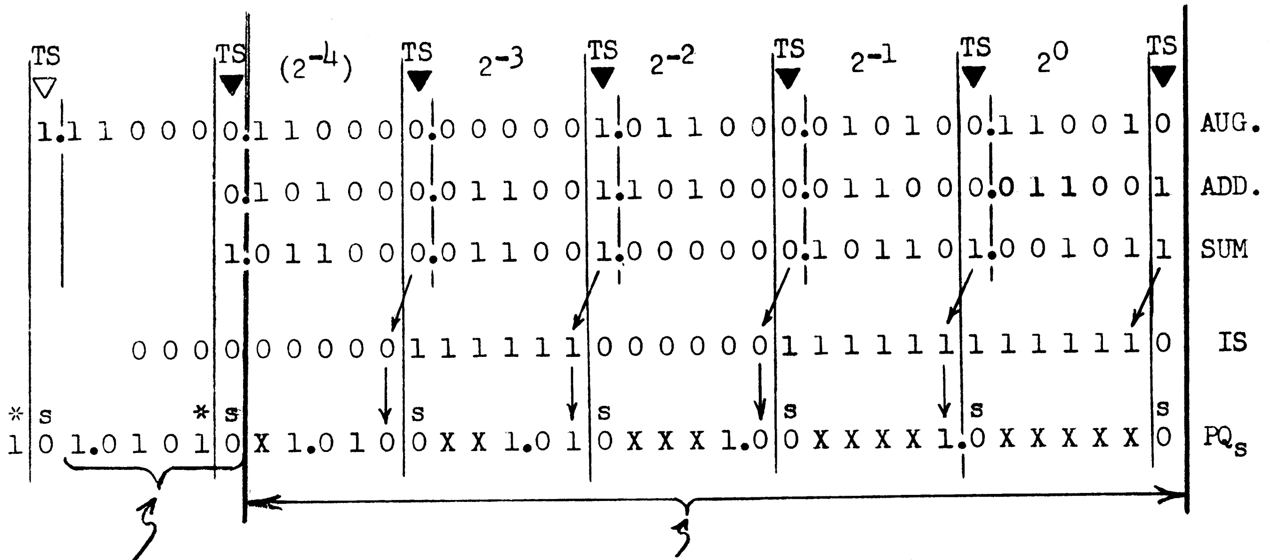
			q			
	d		1.0100			
(C)	.10100● /		.11001●	n	$0 \leq (n = r_0) < 1$	
		sub	.10100●	d	$\frac{1}{2} \leq d < 1$	
	1	+	.00101●	$r_1 = (n - d(2^0))2^0$	$-1 < r_1 < \frac{1}{2}$	*
	•	+	0.01010●	$2r_1 = (n - d(2^0))2^1$	$-2 < 2r_1 < 1$	
		sub	.10100●	d	$\frac{1}{2} \leq d < 1$	
	0	-	.01010●	$r_2 = (n - d(2^0+2^{-1}))2^1$	$-1 < r_2 < \frac{1}{2}$	*
		-	0.10100●	$2r_2 = (n - d(2^0+2^{-1}))2^2$	$-2 < 2r_2 < 1$	
		add	.10100●	d	$\frac{1}{2} \leq d < 1$	
	1	+	.00000●	$r_3 = (n - d(2^0+2^{-2}))2^2$	$-1 < r_3 < \frac{1}{2}$	*
		+	0.00000●	$2r_3 = (n - d(2^0+2^{-2}))2^3$	$-2 < 2r_3 < 1$	
		sub	.10100●	d	$\frac{1}{2} \leq d < 1$	
	0	-	.10100●	$r_4 = (n - d(2^0+2^{-2}+2^{-3}))2^3$	$-1 < r_4 < \frac{1}{2}$	*
		-	1.01000●	$2r_4 = (n - d(2^0+2^{-2}+2^{-3}))2^4$	$-2 < 2r_4 < 1$	
		add	.10100●	d	$\frac{1}{2} \leq d < 1$	
	0	-	.10100●	$r_5 = (n - d(2^0+2^{-2}+2^{-4}))2^4$	$-1 < r_5 < \frac{1}{2}$	*
		-	1.01000●	$2r_5 = (n - d(2^0+2^{-2}+2^{-4}))2^5$	$-2 < 2r_5 < 1$	

* case (B): $|r_n| < |d_{n-1}|$ until $r_n = 0$; thereafter $|r_n| = |d_{n-1}|$

case (C): $|r_n| < |d|$ until $r_n = 0$; thereafter $|r_n| = |d|$

```
                 (D)                              (E)                              (F)

                   s | q                              q                                 q *
 s |  d            0 | 1.0100              d          1.0100                d           1.0101
 0 | 0.10100 / N   0 | 0.11001  n   0.10100 / N   0.11001  n    0.10100/ N   0.11001  n }
           add C   1 | 1.01100              add C   1.01100            add C'  0.01100    } 2^0      1
               N   0 | 0.00101  +               N   0.00101  +             N'  1.00101  + }          .
               N   0 | 0.01010                  N   0.01010                N   0.01010    }
           add C   1 | 1.01100              add C   1.01100            add C'  0.01100    } 2^-1     0
               C   1 | 1.10110  -               C   1.10110  -             C'  0.10110  - }
               C   1 | 1.01100                  C   1.01100                C   1.01100    }
           add N   0 | 0.10100              add N   0.10100            add N'  1.10100    } 2^-2     1
               N   0 | 0.00000  +               N   0.00000  +             N'  1.00000  + }
               N   0 | 0.00000                  N   0.00000                N   0.00000    }
           add C   1 | 1.01100              add C   1.01100            add C'  0.01100    } 2^-3     0
               C   1 | 1.01100  -               C   1.01100  -             C'  0.01100  - }
               C   1 | 0.11000                "N"   0.11000              "N"   0.11000    }
           add N   0 | 0.10100              add N   0.10100            add N   0.10100    } (2^-4)   *
               C   1 | 1.01100  -               C   1.01100  -             C   1.01100  - } round-off 1
               C   1 | 0.11000                "N"   0.11000              "C"   1.11000
```

(G)

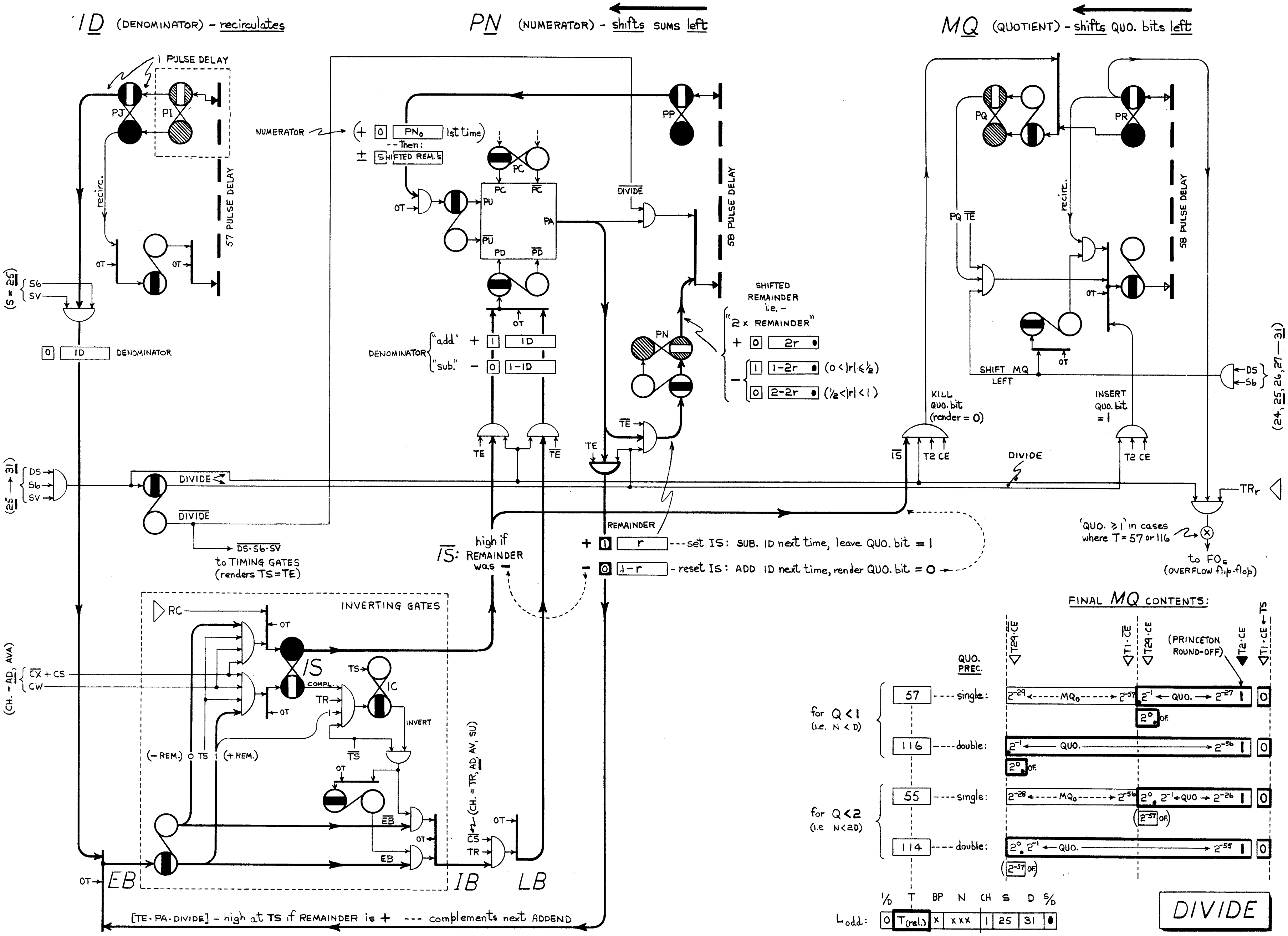


```
TS           TS            TS            TS            TS            TS            TS
▽            ▼  (2^-4)     ▼   2^-3      ▼   2^-2      ▼   2^-1      ▼   2^0       ▼
1.1 1 0 0 0 | 0.1 1 0 0 0 | 0.0 0 0 0 0 | 1.0 1 1 0 0 | 0.0 1 0 1 0 | 0.1 1 0 0 1 | 0   AUG.
            | 0.1 0 1 0 0 | 0.0 1 1 0 0 | 1.1 0 1 0 0 | 0.0 1 1 0 0 | 0.0 1 1 0 0 | 1   ADD.
            | 1.0 1 1 0 0 | 0.0 1 1 0 0 | 1.0 0 0 0 0 | 0.1 0 1 1 0 | 1.0 0 1 0 1 | 1   SUM

      0 0 0 | 0 0 0 0 0 0 | 1 1 1 1 1 1 | 0 0 0 0 0 0 | 1 1 1 1 1 1 | 1 1 1 1 1 1 | 0   IS

*|s       * | s           | s           | s           | s           | s           | s
1|0 1.0 1 0 1 | 0 X 1.0 1 0 | 0 X X 1.0 1 | 0 X X X 1.0 | 0 X X X X 1.| 0 X X X X X | 0   PQs
```

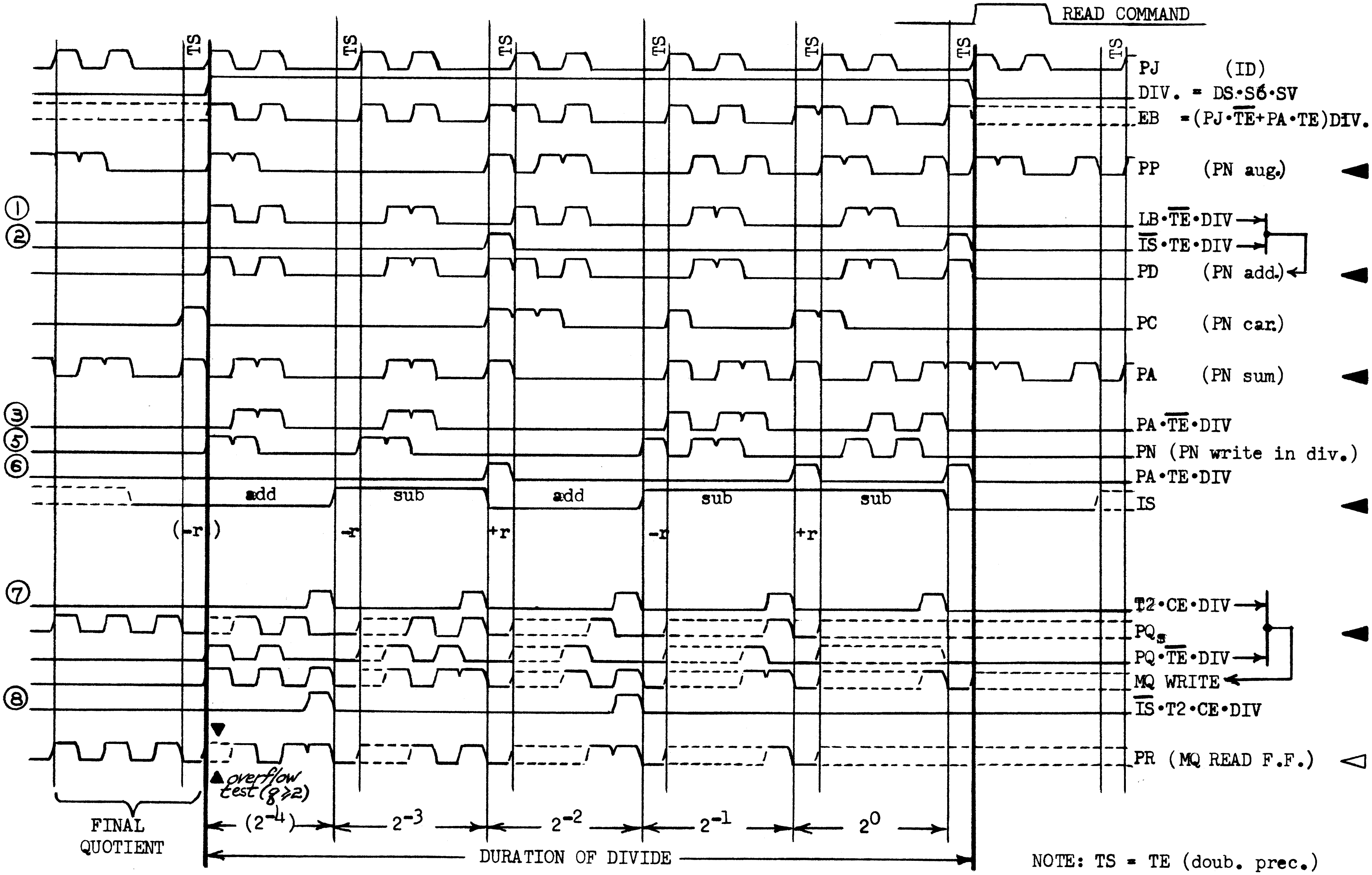
FINAL QUOTIENT (henceforth the recirculating contents of MQ)

DURATION OF DIVIDE (DS•S6•SV)

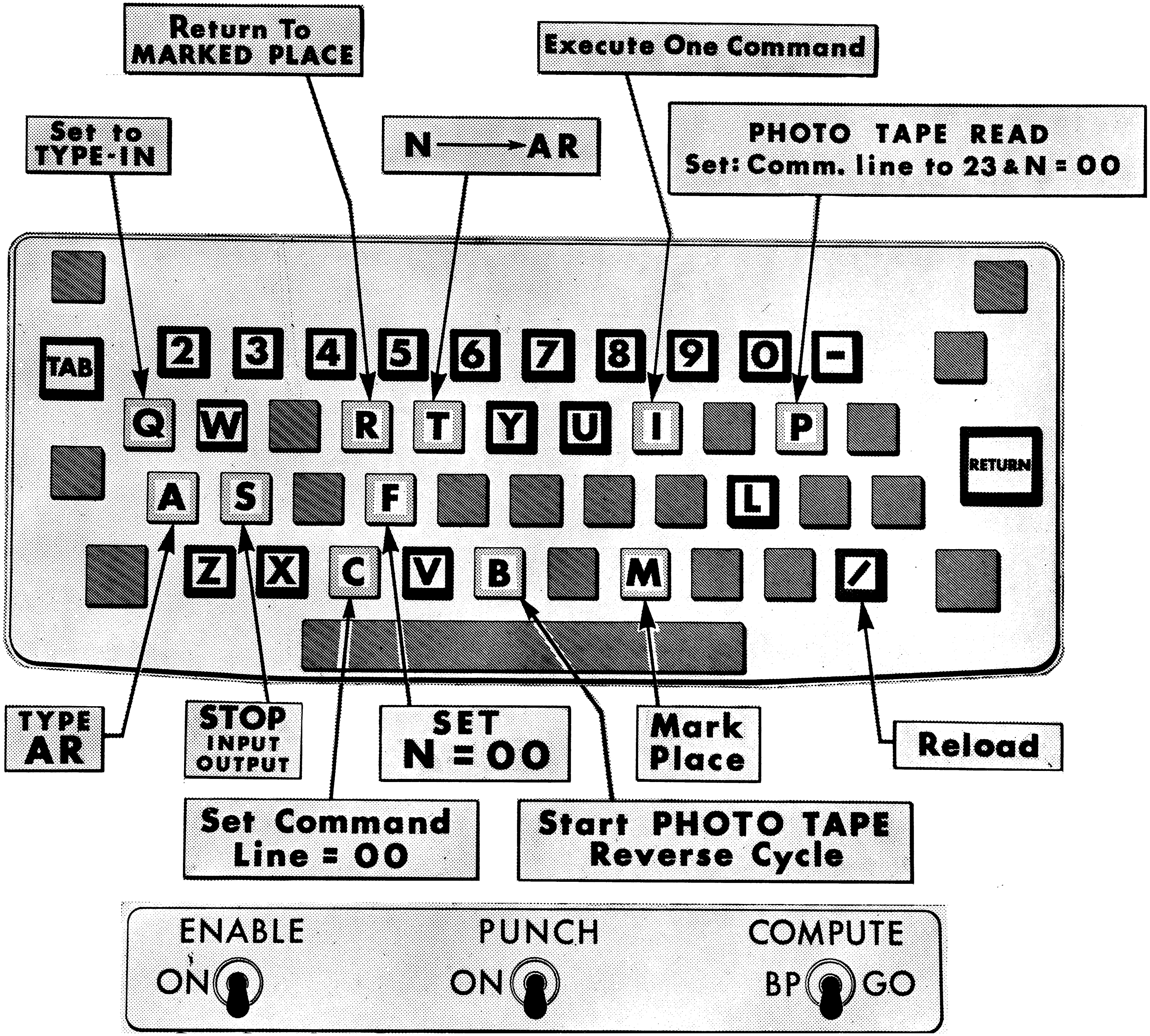
* placed over a "1" means that "1" is the result of an automatic PRINCETON ROUNDOFF. (i.e. lowest order quotient bit is unconditionally equal to 1 regardless of the sign of the remainder previously generated)

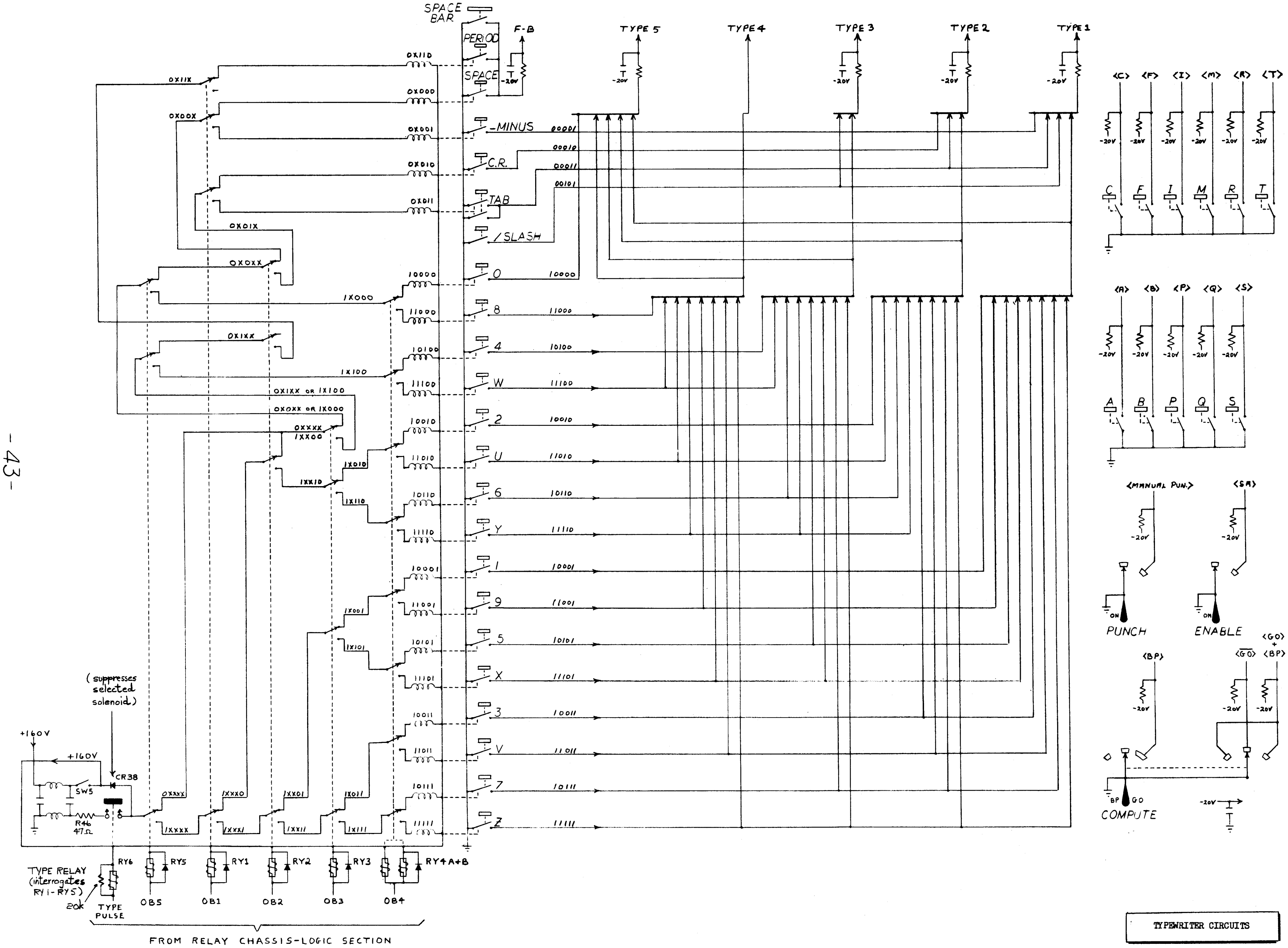

ID (DENOMINATOR) – recirculates
PN (NUMERATOR) – shifts SUMS left
MQ (QUOTIENT) – shifts QUO. bits left
1 PULSE DELAY
PJ
PI
recirc.
57 PULSE DELAY
OT
(S = 25) { S6 SV
0 ID DENOMINATOR
NUMERATOR
+ 0 PN₀ 1st time)
– then:
± SHIFTED REM.'S
PC
PU
PA
PD
OT
DENOMINATOR { "add" + 1 ID
"sub." – 0 1-ID
TE
PP
58 PULSE DELAY
DIVIDE
SHIFTED REMAINDER i.e. – "2 × REMAINDER"
+ 0 2r
– 1 1-2r (0<|r|≤½)
0 2-2r (½<|r|<1)
PN
(25 — 31) { DS S6 SV
DIVIDE
DIVIDE
DS·S6·SV to TIMING GATES (renders TS=TE)
high if IS: REMAINDER was –
REMAINDER
+ 1 r ---set IS: SUB. ID next time, leave QUO. bit = 1
– 0 1-r - reset IS: ADD ID next time, render QUO. bit = 0
PQ
PR
PQ TE
recirc.
58 PULSE DELAY
SHIFT MQ LEFT
KILL QUO. bit (render = 0)
INSERT QUO. bit = 1
IS
T2 CE
DIVIDE
T2 CE
DS S6
(24, 25, 26, 27 — 31)
TRr
'QUO. ≥ 1' in cases where T = 57 or 116
to FO₆ (OVERFLOW flip-flop)
INVERTING GATES
RC
IS
COMPL.
TS
IC
TR
INVERT
TS
(CH. = AD, AVA) { CX + CS CW
(–REM.) 0 TS 1 (+REM.)
EB
EB
(CH. = TR, AD, AV, SU)
CS
TR
EB
IB
LB
[TE·PA·DIVIDE] – high at TS if REMAINDER is + --- complements next ADDEND
FINAL MQ CONTENTS:
T29·CE
T1·CE
T29·CE
(PRINCETON ROUND-OFF)
T2·CE
T1·CE ← TS
QUO. PREC.
for Q<1 (i.e. N<D)
57 ---- single: 2⁻²⁹ ← --- MQ₀ --- → 2⁻⁵⁷ 2⁻¹ ← QUO. → 2⁻²⁷ 1 0
2⁰ OF
116 ---- double: 2⁻¹ ← QUO. → 2⁻⁵⁶ 1 0
2⁰ OF
for Q<2 (i.e. N<2D)
55 ---- single: 2⁻²⁸ ← --- MQ₀ --- → 2⁻⁵⁶ 2⁰, 2⁻¹ ← QUO → 2⁻²⁶ 1 0
(2⁻⁵⁷ OF)
114 ---- double: 2⁰, 2⁻¹ ← QUO. → 2⁻⁵⁵ 1 0
(2⁻⁵⁷ OF)
½ T BP N CH S D %
L odd: 0 T(rel.) x xxx 1 25 31 •
DIVIDE

DIVIDE EXAMPLE (six bits incl. sign): $.11001\overset{s}{0}$ / $.10100\overset{s}{0}$ = $1.0101\overset{s}{0}$

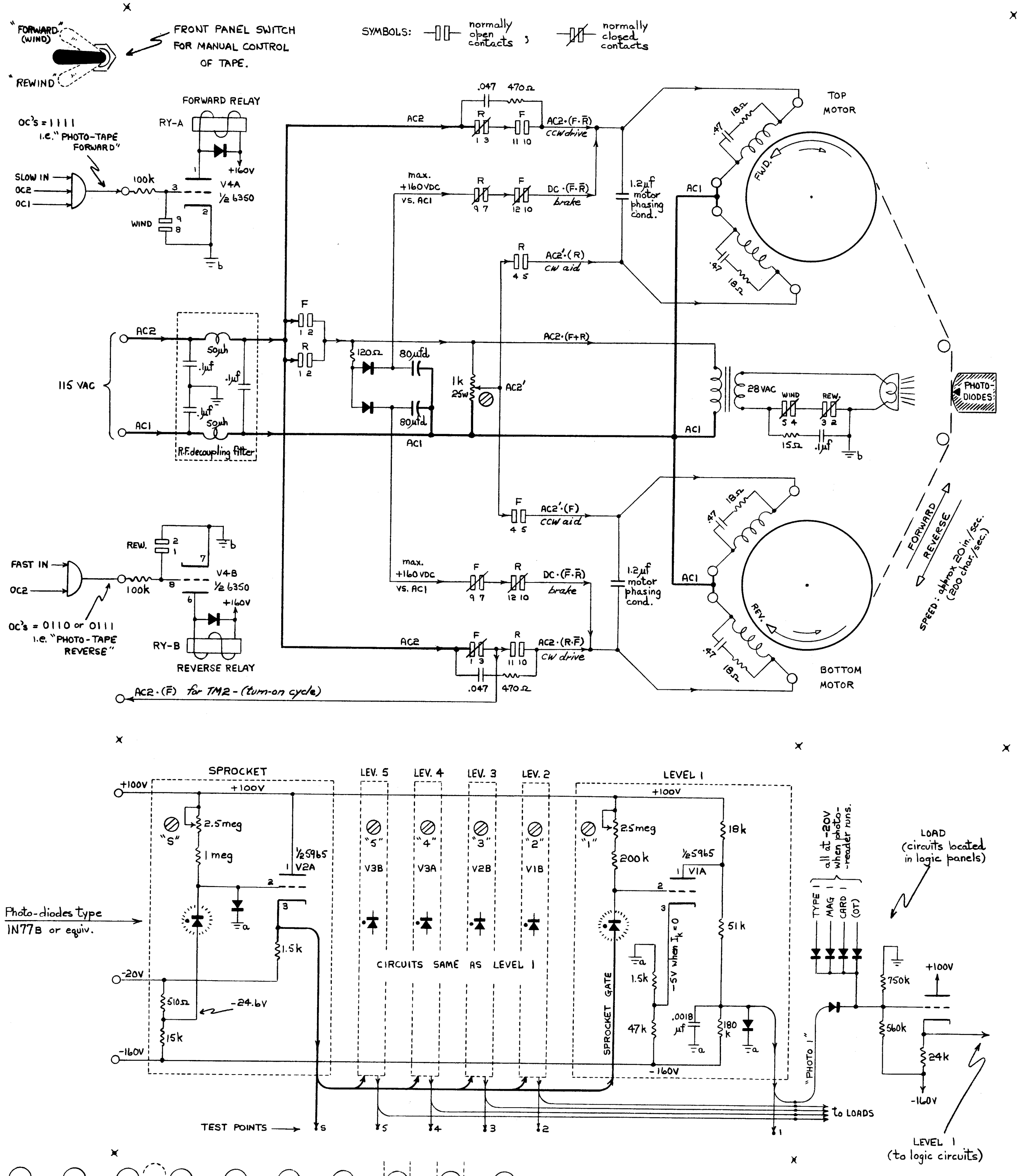


NOTE: TS = TE (doub. prec.)


Return To MARKED PLACE
Execute One Command
Set to TYPE-IN
N → AR
PHOTO TAPE READ
Set: Comm. line to 23 & N = 00
TAB
2
3
4
5
6
7
8
9
0
-
Q
W
R
T
Y
U
I
P
RETURN
A
S
F
L
Z
X
C
V
B
M
/
TYPE AR
STOP INPUT OUTPUT
SET N = 00
Mark Place
Reload
Set Command Line = 00
Start PHOTO TAPE Reverse Cycle
ENABLE
ON
PUNCH
ON
COMPUTE
BP
GO



TYPEWRITER CIRCUITS



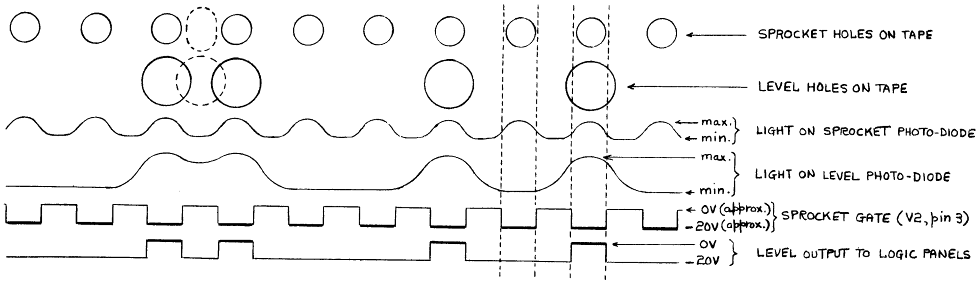
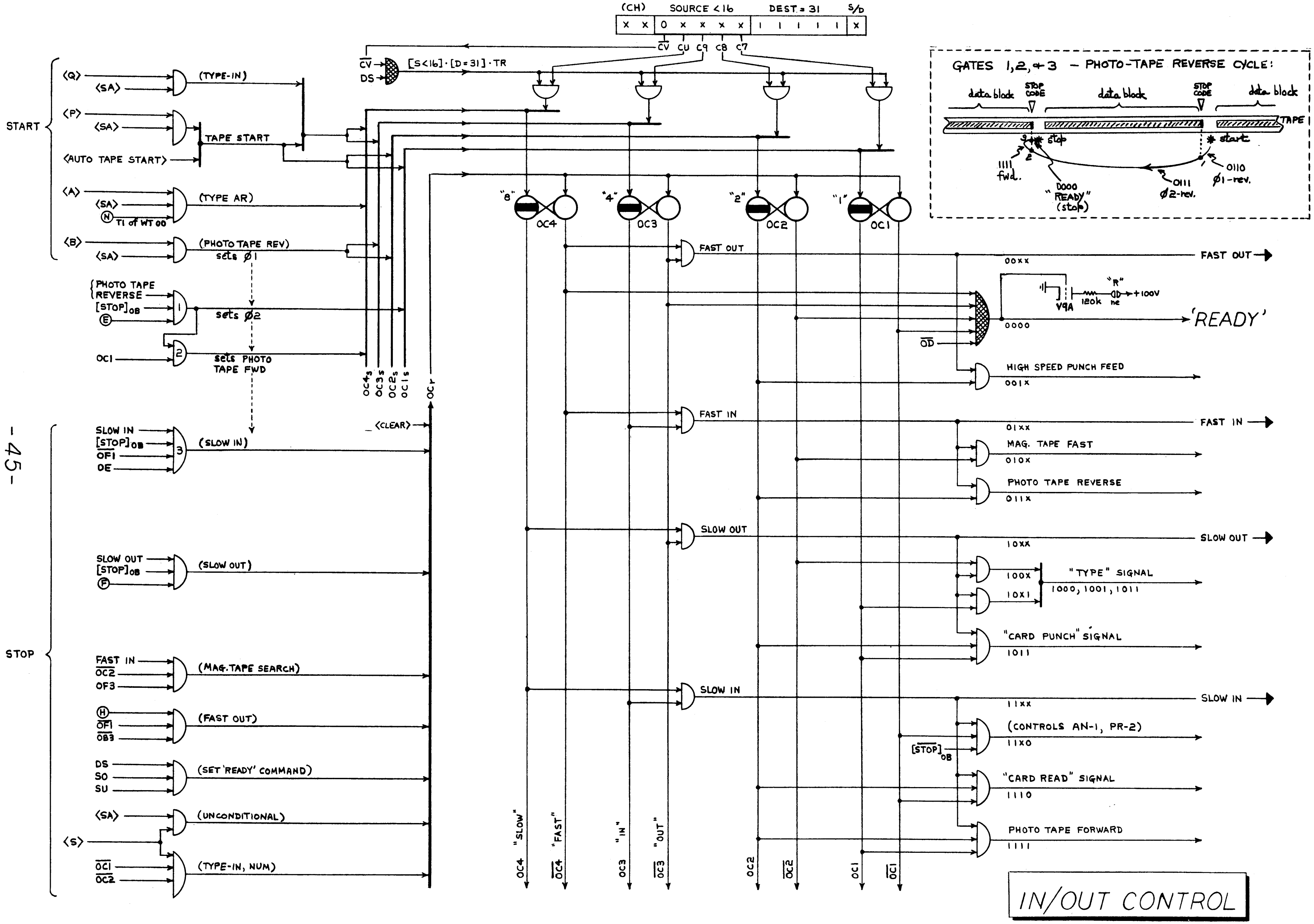

SPROCKET HOLES ON TAPE
LEVEL HOLES ON TAPE
max.
min.
LIGHT ON SPROCKET PHOTO-DIODE
max.
min.
LIGHT ON LEVEL PHOTO-DIODE
0V (approx.)
-20V (approx.)
SPROCKET GATE (V2, pin 3)
0V
-20V
LEVEL OUTPUT TO LOGIC PANELS

PHOTO-TAPE READER

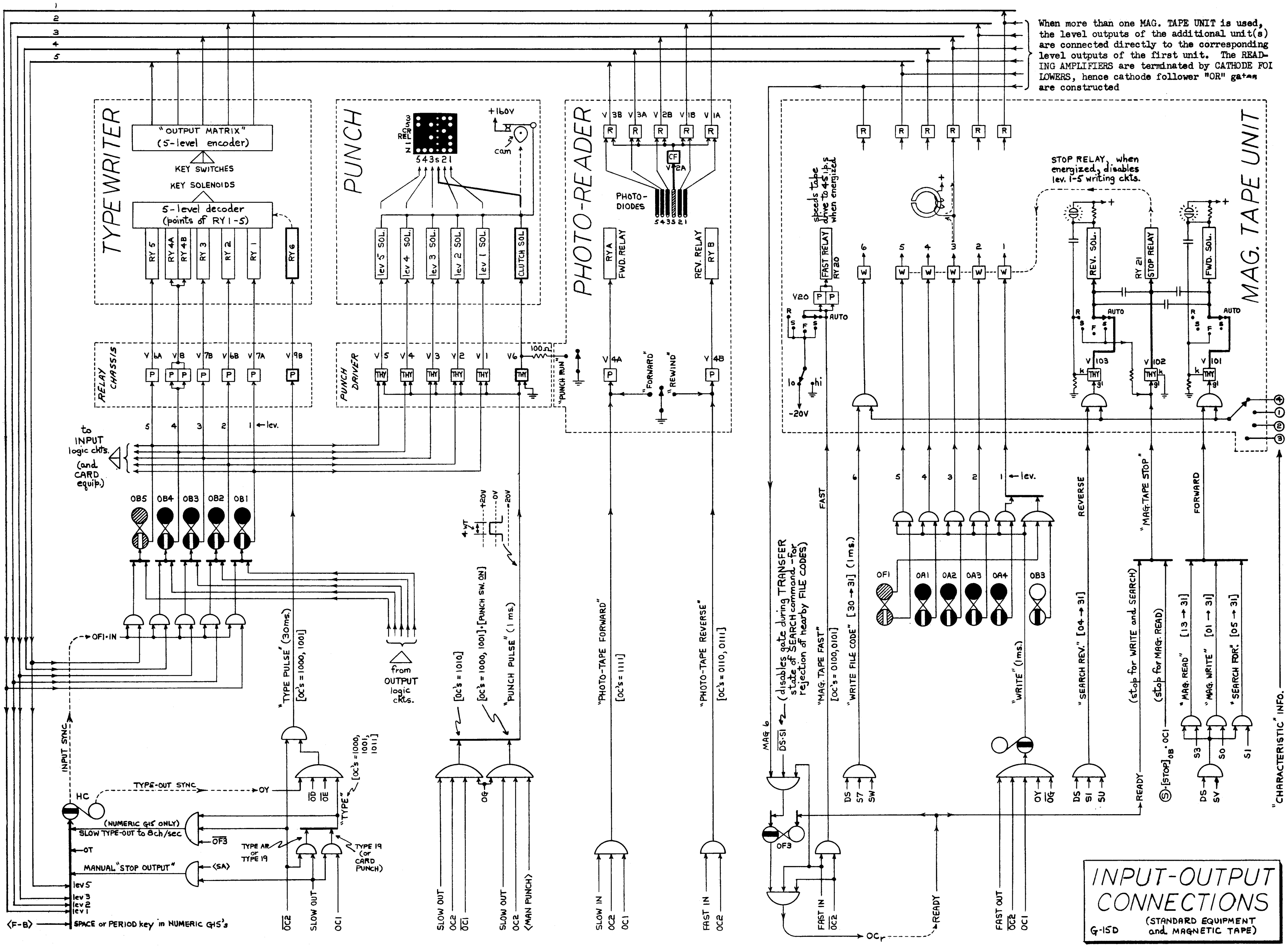
G15D



IN/OUT CONTROL

INPUT-OUTPUT CODES

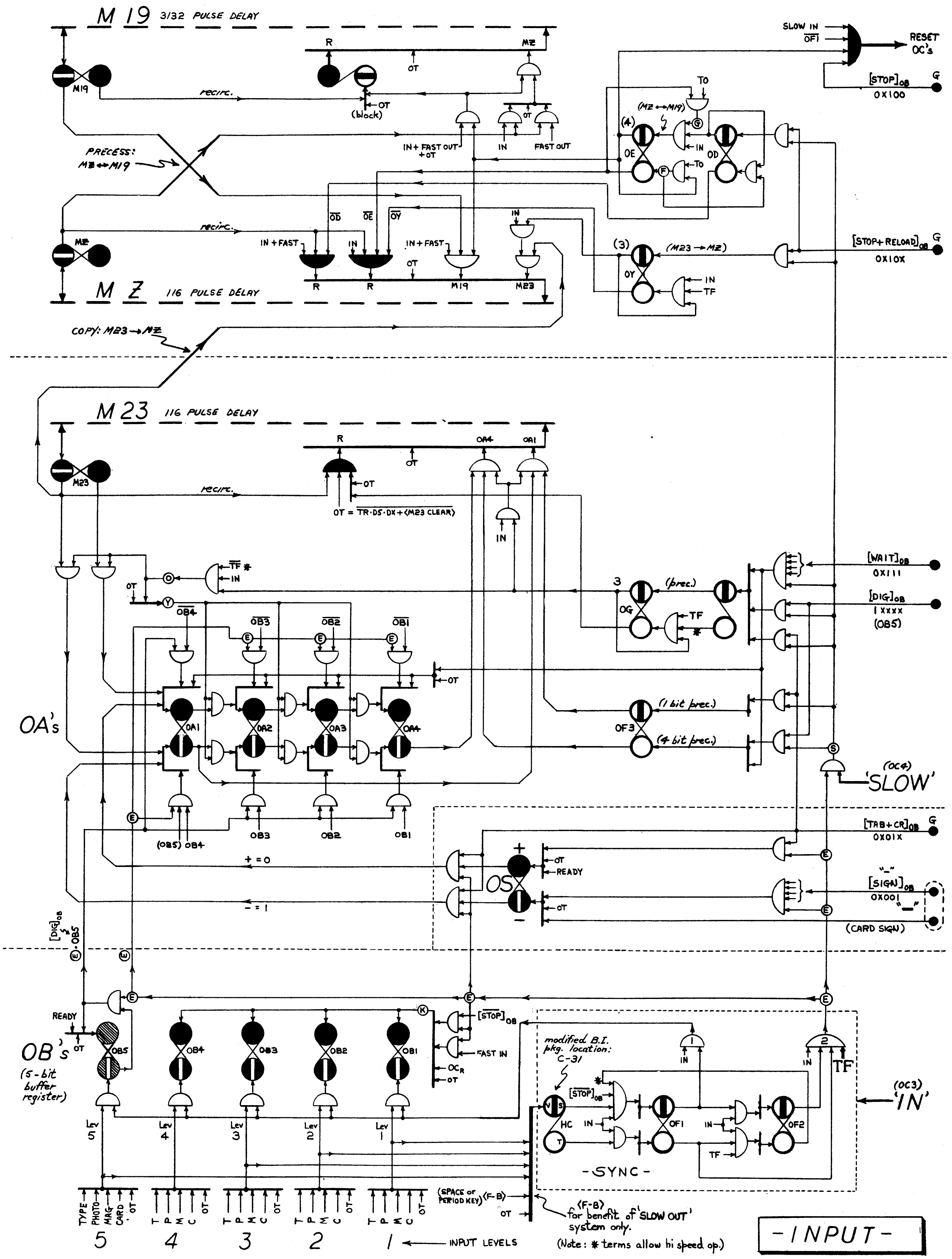
		INPUT			OUTPUT			IN-OUT
CHARACTER or FUNCTION	CODE lev: 5 4 3 2 1	RE-ACTION	ORIGIN		TO SLO OUTPUT	REACTION		Mag. or hi. sp. punch
			Paper tape	Type-writer		Paper tape	Type-writer	
0	1 0 0 0 0	x	x	x	x	x	x	x
1	1 0 0 0 1	x	x	x	x	x	x	x
2	1 0 0 1 0	x	x	x	x	x	x	x
3	1 0 0 1 1	x	x	x	x	x	x	x
4	1 0 1 0 0	x	x	x	x	x	x	x
5	1 0 1 0 1	x	x	x	x	x	x	x
6	1 0 1 1 0	x	x	x	x	x	x	x
7	1 0 1 1 1	x	x	x	x	x	x	x
8	1 1 0 0 0	x	x	x	x	x	x	x
9	1 1 0 0 1	x	x	x	x	x	x	x
u	1 1 0 1 0	x	x	x	x	x	x	x
v	1 1 0 1 1	x	x	x	x	x	x	x
w	1 1 1 0 0	x	x	x	x	x	x	x
x	1 1 1 0 1	x	x	x	x	x	x	x
y	1 1 1 1 0	x	x	x	x	x	x	x
z	1 1 1 1 1	x	x	x	x	x	x	x
Space	0 0 0 0 0		(x)		x	(x)	x	
Minus	0 0 0 0 1	x	x	x	x	x	x	x
CR	0 0 0 1 0	x	x	x	x	x	x	
Tab	0 0 0 1 1	x	x	x	x	x	x	
End (Stop)	0 0 1 0 0	x	x		x	x		x
Reload /	0 0 1 0 1	x	x	x	x	x		x
Period	0 0 1 1 0		x		x	x	x	
Wait	0 0 1 1 1	x	x		x	x		



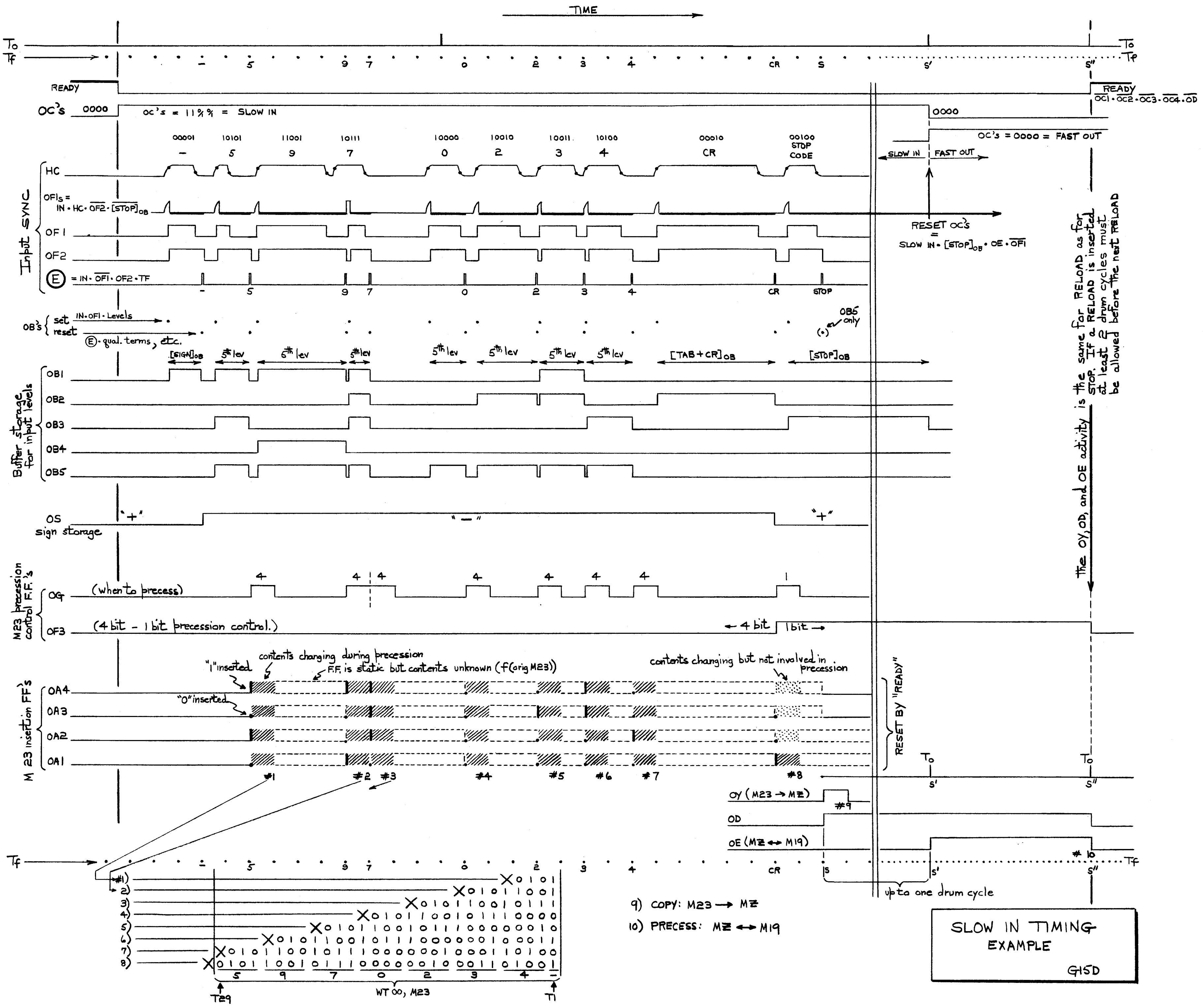
When more than one MAG. TAPE UNIT is used, the level outputs of the additional unit(s) are connected directly to the corresponding level outputs of the first unit. The READING AMPLIFIERS are terminated by CATHODE FOLLOWERS, hence cathode follower "OR" gates are constructed

INPUT-OUTPUT CONNECTIONS

G-15D (STANDARD EQUIPMENT and MAGNETIC TAPE)


M 19 3/32 PULSE DELAY
SLOW IN
OF1
RESET OC's
[STOP]OB
0X100
G
recirc.
M19
R
MZ
OT
OT (block)
IN+ FAST OUT +OT
IN
FAST OUT
TO
(MZ↔M19)
(4)
OE
OD
IN
TO
G
F
PRECESS: MZ↔M19
OD
OE
OY
IN
IN + FAST
IN
IN+ FAST
recirc.
MZ
R
R
M19
M23
(3)
(M23→MZ)
OY
IN
TF
[STOP+ RELOAD]OB
0X10X
M Z 116 PULSE DELAY
COPY: M23→MZ
M 23 116 PULSE DELAY
R
OA4
OA1
OT
M23
recirc.
OT
OT = TR·DS·DX+(M23 CLEAR)
IN
TF *
IN
O
OT
Y
OB4
OB3
OB2
OB1
E
3
(prec.)
OG
TF
*
[WAIT]OB
0X111
[DIG]OB
1XXXX
(OB5)
OT
OA's
OA1
OA2
OA3
OA4
OF3
(1 bit prec.)
(4 bit prec.)
S
(OC4)
'SLOW'
(OB5) OB4
OB3
OB2
OB1
[TAB+CR]OB
0X01X
+ = 0
OS
+
OT
READY
OT
−
− = 1
"−"
[SIGN]OB
0X001
"—"
(CARD SIGN)
[DIG]OB · E · OB5
E
READY
OT
OB's (5-bit buffer register)
OB5
OB4
OB3
OB2
OB1
K
[STOP]OB
FAST IN
OC_R
OT
modified B.I. pkg. location: C-31
1
IN
2
IN
TF
(OC3) 'IN'
V
S
HC
T
[STOP]OB
*
IN
OF1
IN
OF2
TF
-SYNC-
Lev 5
Lev 4
Lev 3
Lev 2
Lev 1
(SPACE or PERIOD KEY) (F-B)
OT
TYPE
PHOTO
MAG
CARD
OT
T P M C OT
(F-B) for benefit of 'SLOW OUT' system only.
(Note: * terms allow hi speed op.)
5 4 3 2 1 ← INPUT LEVELS

- INPUT -


TIME
To
Tf
- 5 9 7 0 2 3 4 CR S S' S''
READY
OC's 0000
OC's = 11%% = SLOW IN
0000
OC's = 0000 = FAST OUT
READY
$\overline{OC1}\cdot\overline{OC2}\cdot\overline{OC3}\cdot\overline{OC4}\cdot\overline{OD}$
00001 10101 11001 10111 10000 10010 10011 10100 00010 00100
- 5 9 7 0 2 3 4 CR
STOP CODE
SLOW IN
FAST OUT
Input SYNC
HC
$OF1_S = IN \cdot HC \cdot \overline{OF2} \cdot [\overline{STOP}]_{OB}$
OF1
OF2
Ⓔ = $IN \cdot \overline{OF1} \cdot OF2 \cdot TF$
- 5 9 7 0 2 3 4 CR STOP
RESET OC's = SLOW IN $\cdot [STOP]_{OB} \cdot OE \cdot \overline{OF1}$
OB's { set IN·OF1·Levels
reset Ⓔ·qual. terms, etc.
OB5 only (•)
$[SIGN]_{OB}$
5th lev
5th lev
5th lev
5th lev
5th lev
5th lev
5th lev
$[TAB+CR]_{OB}$
$[STOP]_{OB}$
Buffer storage for input levels
OB1
OB2
OB3
OB4
OB5
OS sign storage
"+"
"−"
"+"
M23 precession control F.F.'s
OG
(when to precess)
4 4 4 4 4 4 4 1
OF3
(4 bit - 1 bit precession control.)
← 4 bit
1 bit →
M 23 insertion FF's
OA4
OA3
OA2
OA1
"1" inserted
"0" inserted
contents changing during precession
F.F. is static but contents unknown (f(orig M23))
contents changing but not involved in precession
#1 #2 #3 #4 #5 #6 #7 #8
RESET BY "READY"
To S'
To S''
OY (M23 → MZ)
#9
OD
OE (MZ ↔ M19)
#10
Tf
- 5 9 7 0 2 3 4 CR S S' S''
up to one drum cycle
the OY, OD, and OE activity is the same for RELOAD as for STOP. If a RELOAD is inserted at least 2 drum cycles must be allowed before the next RELOAD
#1) X0101
2) X0101 1001
3) X0101 1001 0111
4) X0101 1001 0111 0000
5) X0101 1001 0111 0000 0010
6) X0101 1001 0111 0000 0010 0011
7) X0101 1001 0111 0000 0010 0011 0100
8) X0101 1001 0111 0000 0010 0011 0100 1
5 9 7 0 2 3 4 −
T29
WT 00, M23
T1

9) COPY: M23 → MZ

10) PRECESS: MZ ↔ M19

SLOW IN TIMING
EXAMPLE

G15D

SLOW OUT FORMAT

Format Characters

Character	Symbol	Code
Digit	D	000
Sign	S	100
Carriage ret.	CR	010
Tab	T	110
End (Stop)	E	001
Reload	R	101
Period	P	011
Wait	W	111

Standard Formats

AR to output: S D D D D D D D CR E

```
 S   D   D   D   D   D   D   D   CR  E
100 000 000 000 000 000 000 000 010 001 0-----      -----0
```

Hex. for input: 8000004 8000000 0000000 0000000 /s

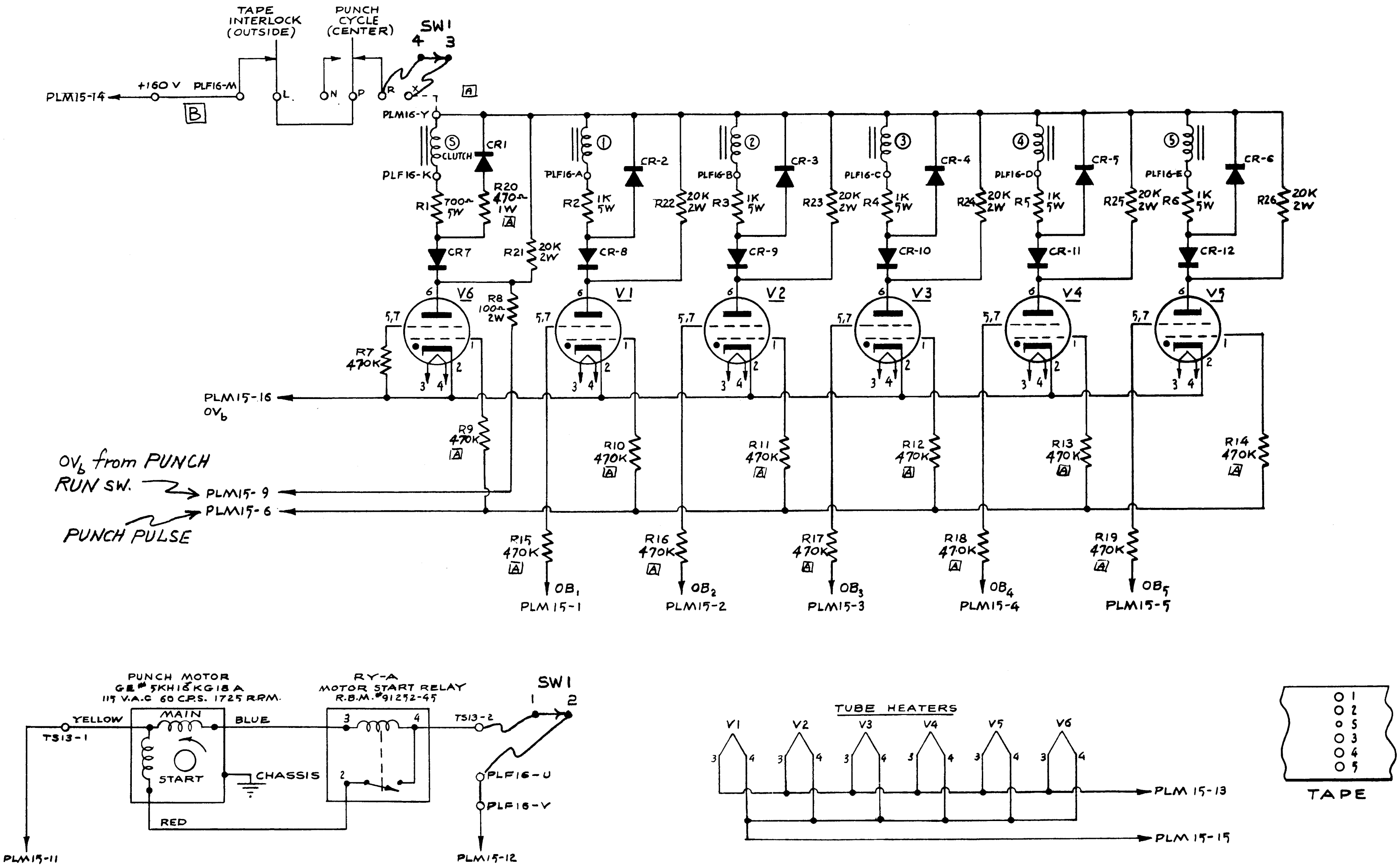
Words 0 - 3 of Line 3 are used for storage of AR format. Highest order format bit should be in T29 of Word 3.

LINE 19 to output: SDDDDDDDTSDDDDDDDTSDDDDDDDTSDDDDDDD (CR) E

Hex. for input: 800000x 0000034 00000x0 0000110 /s

Words 0 - 3 of Line 2 are used for storage of Line 19 format. Highest order format bit should be T29 of Word 3.

Maximum number of format characters per format = 38



5. CR7 THRU. CR12 INCL. - "FEDERAL" #1101A OR EQUIV.

4. CR1 THRU. CR6 : CR-38 - "INTERNATIONAL RECTIFIER CORP."

3. MOTOR ROTATION VIEWED FROM END OPPOSITE SHAFT EXTENSION

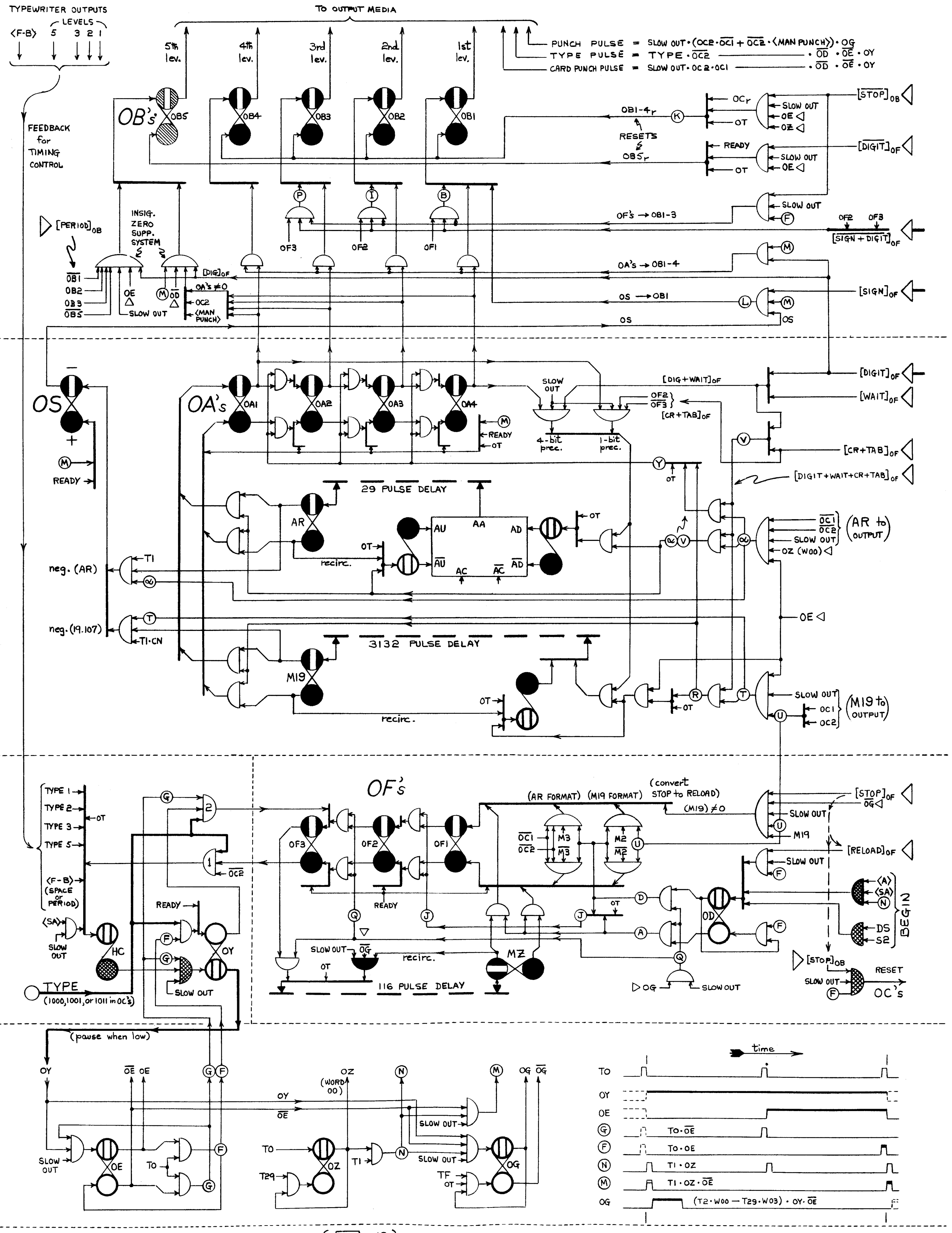
2. ALL RESISTORS: 1/2 W. ± 5%

1. ALL TUBES : 2D21

NOTES: UNLESS OTHERWISE SPECIFIED-

3C246

TAPE PUNCH DRIVER

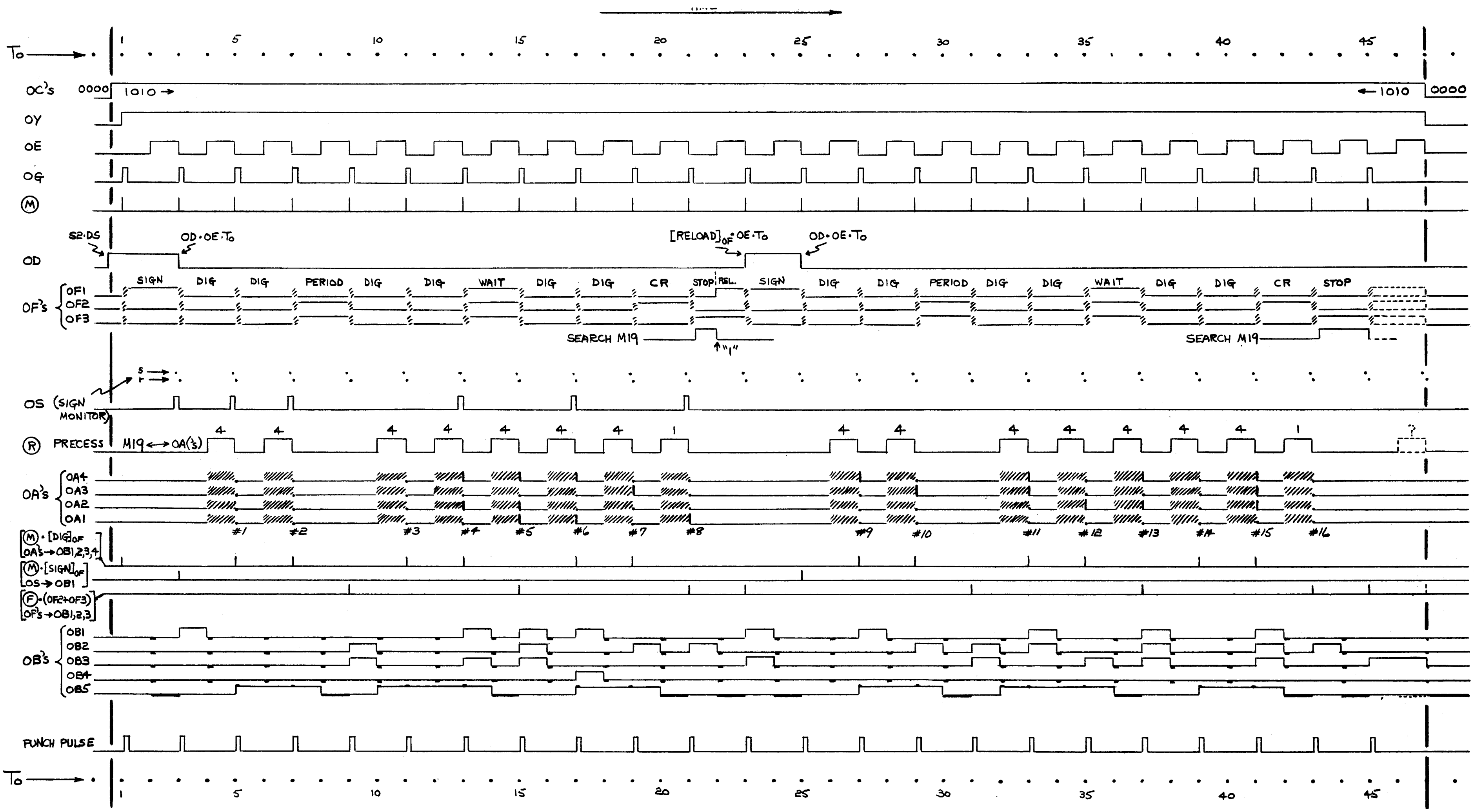


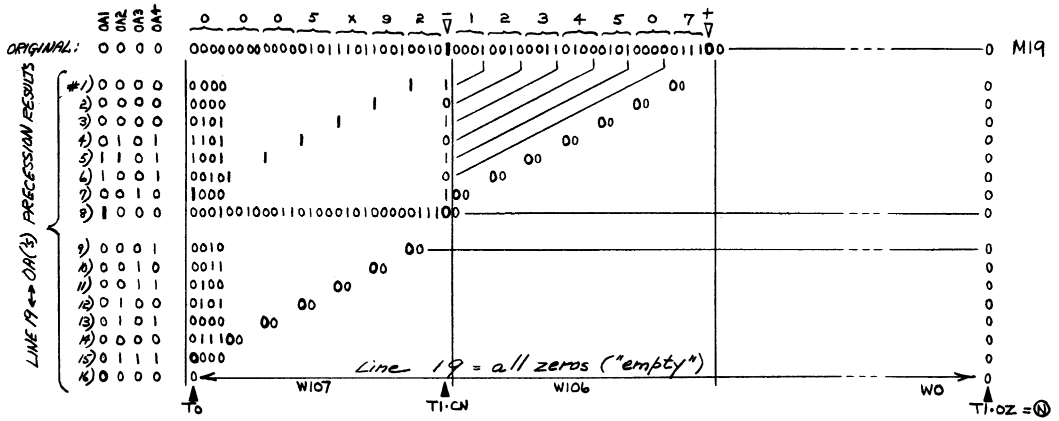
Color key: □ TIMING (□ FEEDBACK) □ FORMAT { □ M19 □ AR } □ DATA EXTRACTION □ OUTPUT BUFFER

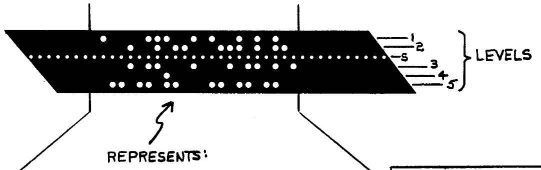
SLOW OUT

FORMAT CHARACTER	OF3	OF2	OF1	Ⓟ	Ⓘ	Ⓑ	OB5	OB4	OB3	OB2	OB1	L19 PRECESSION			TYPE	FUNCTION
												4 BIT	1 BIT	NONE		
DIGIT 000	0	0	0				1/0	0A1	0A2	0A3	0A4	YES			YES	L19↔OAs→OBs→OUTPUT
SIGN 100	0	0	1				0	0	0	0	0S			YES	YES	OS→OB1→OUTPUT
CARRIAGE RETURN 010	0	1	0		|		0	0	0	1	0		YES		YES	OFs→OBs→OUTPUT
TAB 110	0	1	1		|	|	0	0	0	1	1		YES		YES	OFs→OBs→OUTPUT
END (STOP) 001	1	0	0	|			0	0	1	0	0			YES	NO	L19 = 0? YES: STOP→OUTPUT, RESET OC's NO: CHANGE TO RELOAD
RELOAD 101	1	0	1	|		|	0	0	1	0	1			YES	NO	1) OFs→OBs→OUTPUT 2) RELOAD FORMAT
PERIOD 011	1	1	0	|	|		0	0	1	1	0			YES	YES	1) OFs→OBs→OUTPUT 2) SET OB5
WAIT 111	1	1	1	|	|	|	0	0	1	1	1	YES			NO	OFs→OBs→OUTPUT

SLOW OUT







REPRESENTS:

SP – 0 0.0 5 WT 9 2 CR / SP 1 2.3 4 WT 0 7 CR E

(if typed: – 0 0 . 0 5 9 2
1 2 . 3 4 0 7)

SLOW OUT TIMING

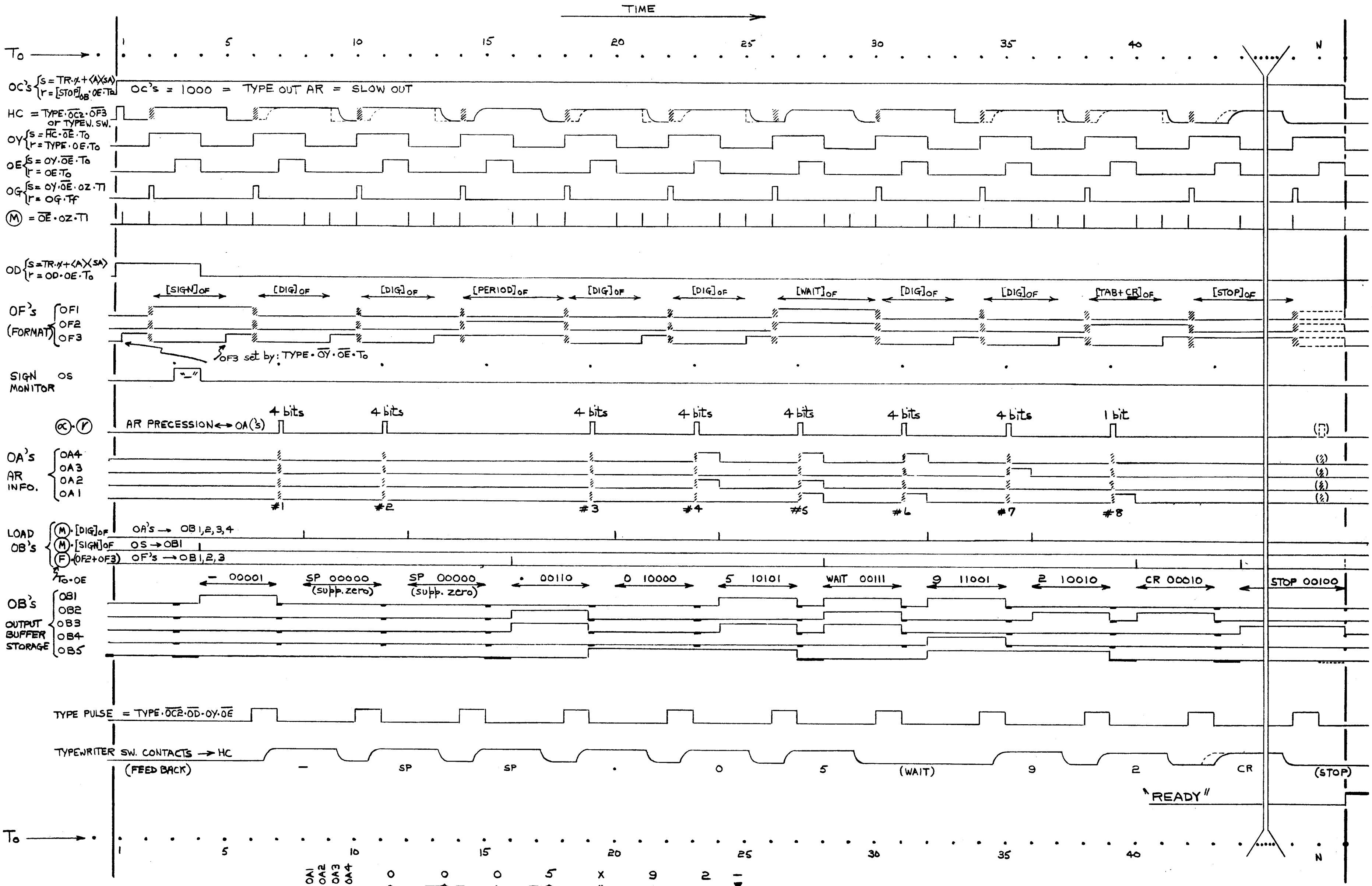
EXAMPLE: PUNCH M19

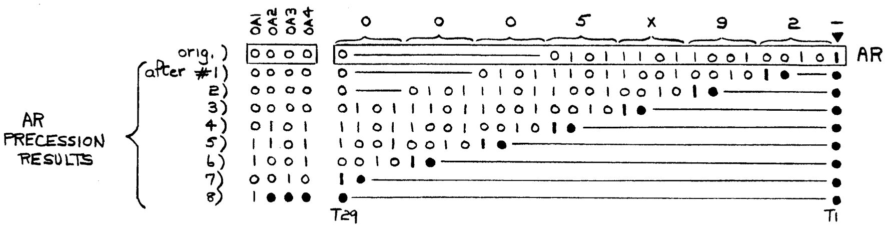
G15D

Format : M2 :

```
 S   D   D   P   D   D   W   D   D   CR  E
100 000 000 011 000 000 111 000 000 010 001 ½₀---
 8   0   3   0   3   8   0   -   1  ///
```

↑ T29 of WT 03


TIME
To
1 5 10 15 20 25 30 35 40 N
OC's {s = TR·α + ⟨A⟩⟨SA⟩; r = [STOP]OB·OE·To
OC's = 1000 = TYPE OUT AR = SLOW OUT
HC = TYPE·OC2·OF3 or TYPEW. SW.
OY {s = HC·OE·To; r = TYPE·OE·To
OE {s = OY·OE·To; r = OE·To
OG {s = OY·OE·OZ·T1; r = OG·Tf
Ⓜ = OE·OZ·T1
OD {s = TR·α + ⟨A⟩⟨SA⟩; r = OD·OE·To
[SIGN]OF [DIG]OF [DIG]OF [PERIOD]OF [DIG]OF [DIG]OF [WAIT]OF [DIG]OF [DIG]OF [TAB+CR]OF [STOP]OF
OF's (FORMAT) OF1 OF2 OF3
OF3 set by: TYPE·OY·OE·To
SIGN MONITOR OS
"–"
α·γ
AR PRECESSION ↔ OA(s)
4 bits 4 bits 4 bits 4 bits 4 bits 4 bits 4 bits 1 bit
OA's AR INFO. OA4 OA3 OA2 OA1
#1 #2 #3 #4 #5 #6 #7 #8
LOAD OB's
Ⓜ·[DIG]OF OA's → OB1,2,3,4
Ⓜ·[SIGN]OF OS → OB1
Ⓕ·(OF2+OF3) OF's → OB1,2,3
To·OE
– 00001
SP 00000 (supp. zero)
SP 00000 (supp. zero)
· 00110
0 10000
5 10101
WAIT 00111
9 11001
2 10010
CR 00010
STOP 00100
OB's OUTPUT BUFFER STORAGE OB1 OB2 OB3 OB4 OB5
TYPE PULSE = TYPE·OC2·OD·OY·OE
TYPEWRITER SW. CONTACTS → HC (FEED BACK)
– SP SP · 0 5 (WAIT) 9 2 CR (STOP)
"READY"



TYPED COPY:

– SPSP . 0592 CR
i.e.:

- | | | . | 0 | 5 | 9 | 2 |

Format in M3:

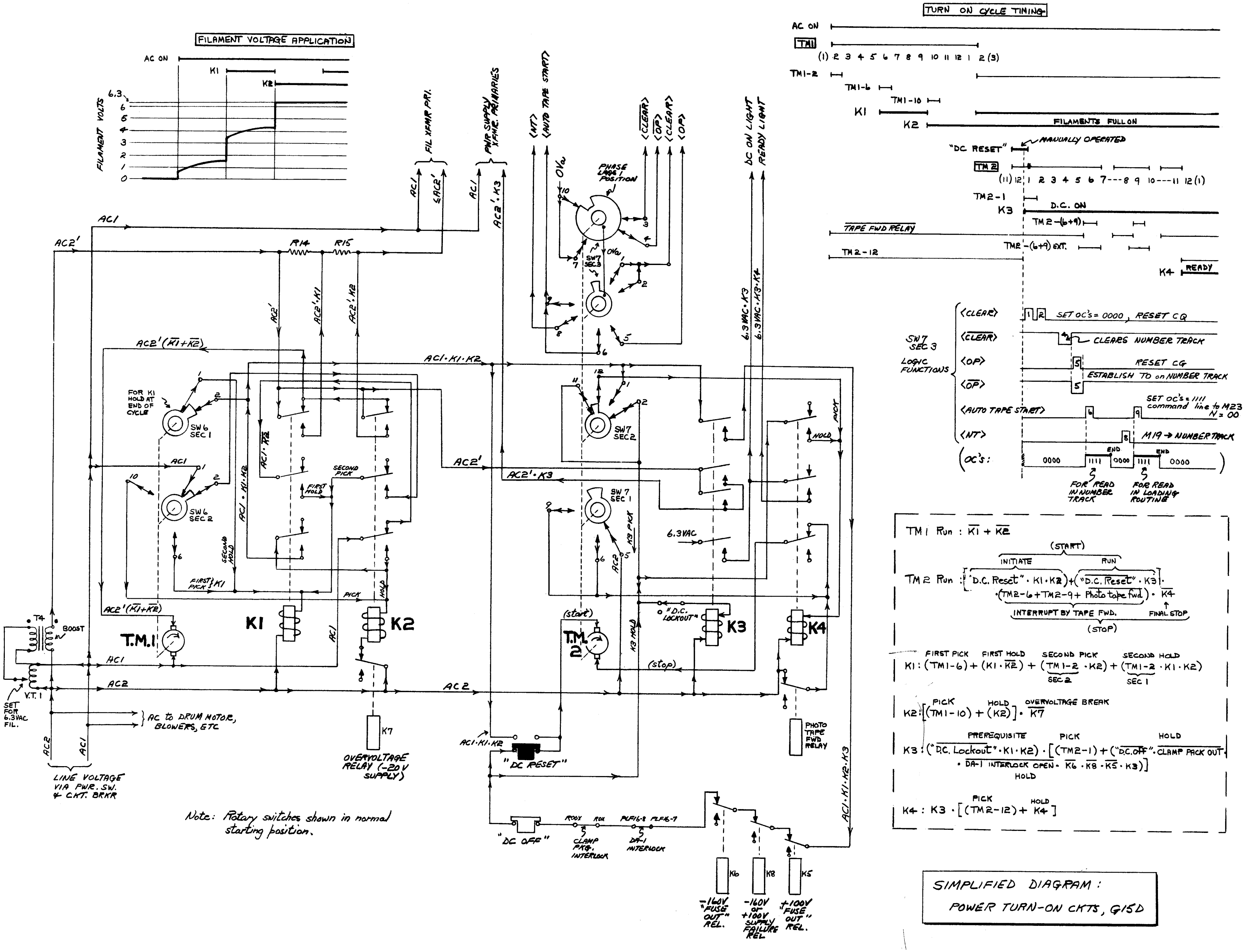
S D D P D D W D D CR E
100 000 000 011 000 000 111 000 000 010 001 ½ –
8 0 3 0 3 8 0 – 1 ///

↑ T29 of WT 03

SLOW OUT TIMING

EXAMPLE: TYPE AR.

G15D



Note: Rotary switches shown in normal starting position.

TM1 Run : $\overline{K1}+\overline{K2}$

$$\text{TM2 Run} : \underbrace{\overset{(\text{START})}{\overbrace{\left[\overset{\text{INITIATE}}{\overbrace{(\text{"D.C. Reset"}\cdot K1\cdot K2)}}+\overset{\text{RUN}}{\overbrace{(\overline{\text{"D.C. Reset"}}\cdot K3}}\right]}}\cdot\underbrace{(\text{TM2-6}+\text{TM2-9}+\overline{\text{Photo tape fwd}})}_{\text{INTERRUPT BY TAPE FWD.}}\cdot\underset{\substack{\uparrow\\ \text{FINAL STOP}}}{\overline{K4}}}_{(\text{STOP})}$$

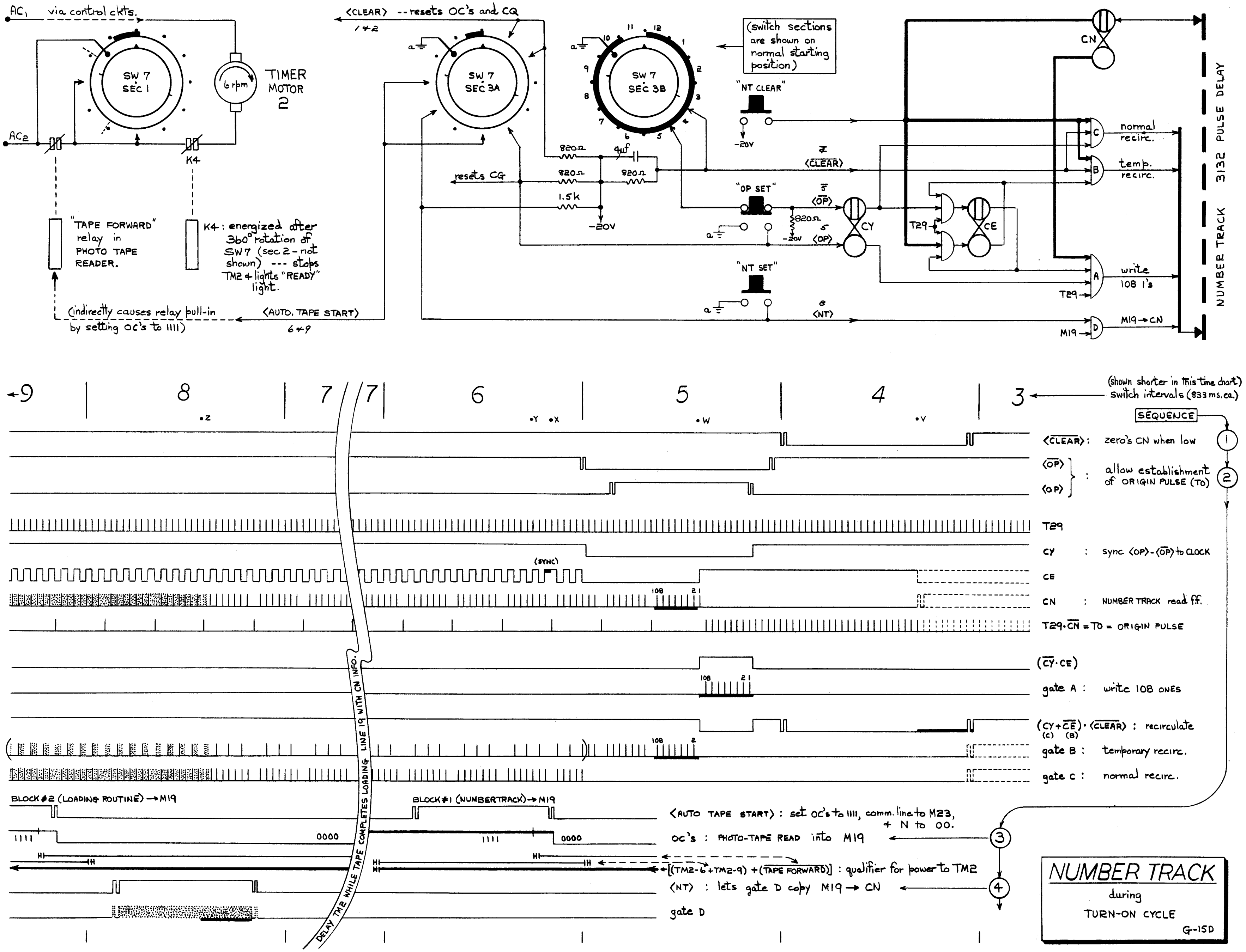
$$K1:\overset{\text{FIRST PICK}}{(\text{TM1-6})}+\overset{\text{FIRST HOLD}}{(K1\cdot\overline{K2})}+\overset{\text{SECOND PICK}}{(\underbrace{\text{TM1-2}}_{\text{SEC 2}}\cdot K2)}+\overset{\text{SECOND HOLD}}{(\underbrace{\text{TM1-2}}_{\text{SEC 1}}\cdot K1\cdot K2)}$$

$$K2:\left[\overset{\text{PICK}}{(\text{TM1-10})}+\overset{\text{HOLD}}{(K2)}\right]\cdot\overset{\text{OVERVOLTAGE BREAK}}{\overline{K7}}$$

$$K3:\overset{\text{PREREQUISITE}}{(\overline{\text{"D.C. Lockout"}}\cdot K1\cdot K2)}\cdot\left[\overset{\text{PICK}}{(\text{TM2-1})}+\overset{\text{HOLD}}{(\text{"}\overline{\text{D.C. OFF}}\text{"}\cdot\overline{\text{CLAMP PACK OUT}}}\cdot\underset{\text{HOLD}}{\overline{\text{DA-1 INTERLOCK OPEN}}\cdot\overline{K6}\cdot\overline{K8}\cdot\overline{K5}\cdot K3)}\right]$$

$$K4:K3\cdot\left[\overset{\text{PICK}}{(\text{TM2-12})}+\overset{\text{HOLD}}{K4}\right]$$

SIMPLIFIED DIAGRAM :
POWER TURN-ON CKTS, G15D

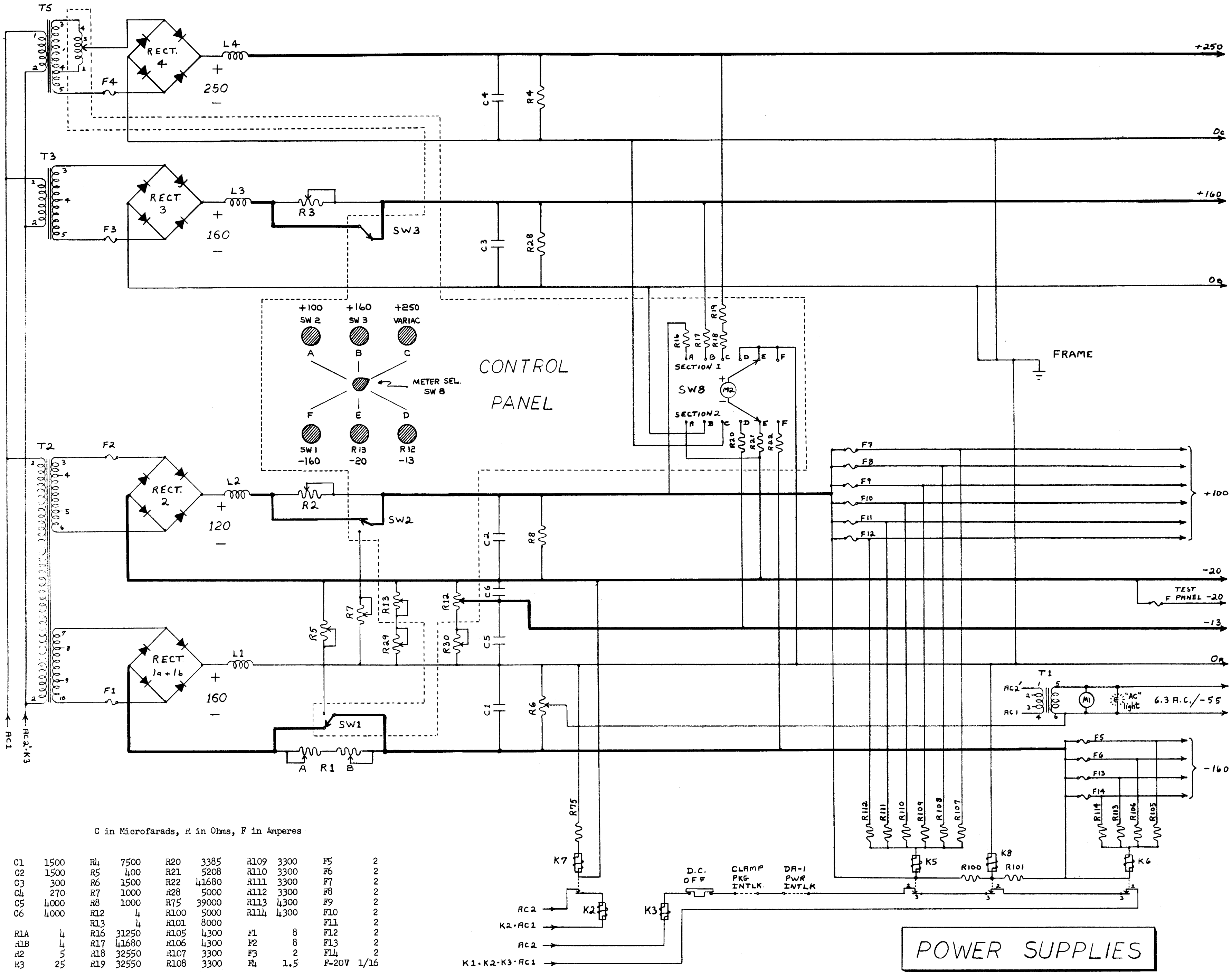

AC₁ via control ckts.
AC₂
SW 7 SEC 1
TIMER MOTOR 2
6 rpm
K4
"TAPE FORWARD" relay in PHOTO TAPE READER.
K4: energized after 360° rotation of SW7 (sec 2 - not shown) --- stops TM2 + lights "READY" light.
(indirectly causes relay pull-in by setting OC's to 1111)
⟨AUTO. TAPE START⟩
6+9
⟨CLEAR⟩ -- resets OC's and CQ
1+2
a
SW 7 SEC 3A
SW 7 SEC 3B
1 2 3 4 5 6 7 8 9 10 11 12
(switch sections are shown on normal starting position)
820Ω
4μf
820Ω
820Ω
1.5k
-20V
resets CG
"NT CLEAR"
-20V
7
⟨CLEAR⟩
"OP SET"
5
⟨OP⟩
820Ω
-20V
5
⟨OP⟩
CY
T29
CE
"NT SET"
8
⟨NT⟩
CN
C
normal recirc.
B
temp. recirc.
A
write 108 1's
T29
D
M19 → CN
M19
3132 PULSE DELAY
NUMBER TRACK
←9 8 7 7 6 5 4 3
(shown shorter in this time chart) switch intervals (833 ms. ea.)
•Z •Y •X •W •V
SEQUENCE
1
2
3
4
⟨CLEAR⟩: zero's CN when low
⟨OP⟩
⟨OP⟩
: allow establishment of ORIGIN PULSE (T0)
T29
CY : sync ⟨OP⟩-⟨OP⟩ to CLOCK
(SYNC)
CE
CN : NUMBER TRACK read ff.
108 2 1
T29·CN = T0 = ORIGIN PULSE
(CY·CE)
gate A : write 108 ONES
108 2 1
(CY+CE)·⟨CLEAR⟩ : recirculate
(C) (B)
gate B : temporary recirc.
108 2
gate C : normal recirc.
BLOCK #2 (LOADING ROUTINE) → M19
BLOCK #1 (NUMBERTRACK) → M19
1111 0000 1111 0000
DELAY TM2 WHILE TAPE COMPLETES LOADING LINE 19 WITH CN INFO.
⟨AUTO TAPE START⟩ : set OC's to 1111, comm. line to M23, + N to 00.
OC's : PHOTO-TAPE READ into M19
[(TM2-6+TM2-9) + (TAPE FORWARD)] : qualifier for power to TM2
⟨NT⟩ : lets gate D copy M19 → CN
gate D

NUMBER TRACK

during

TURN-ON CYCLE

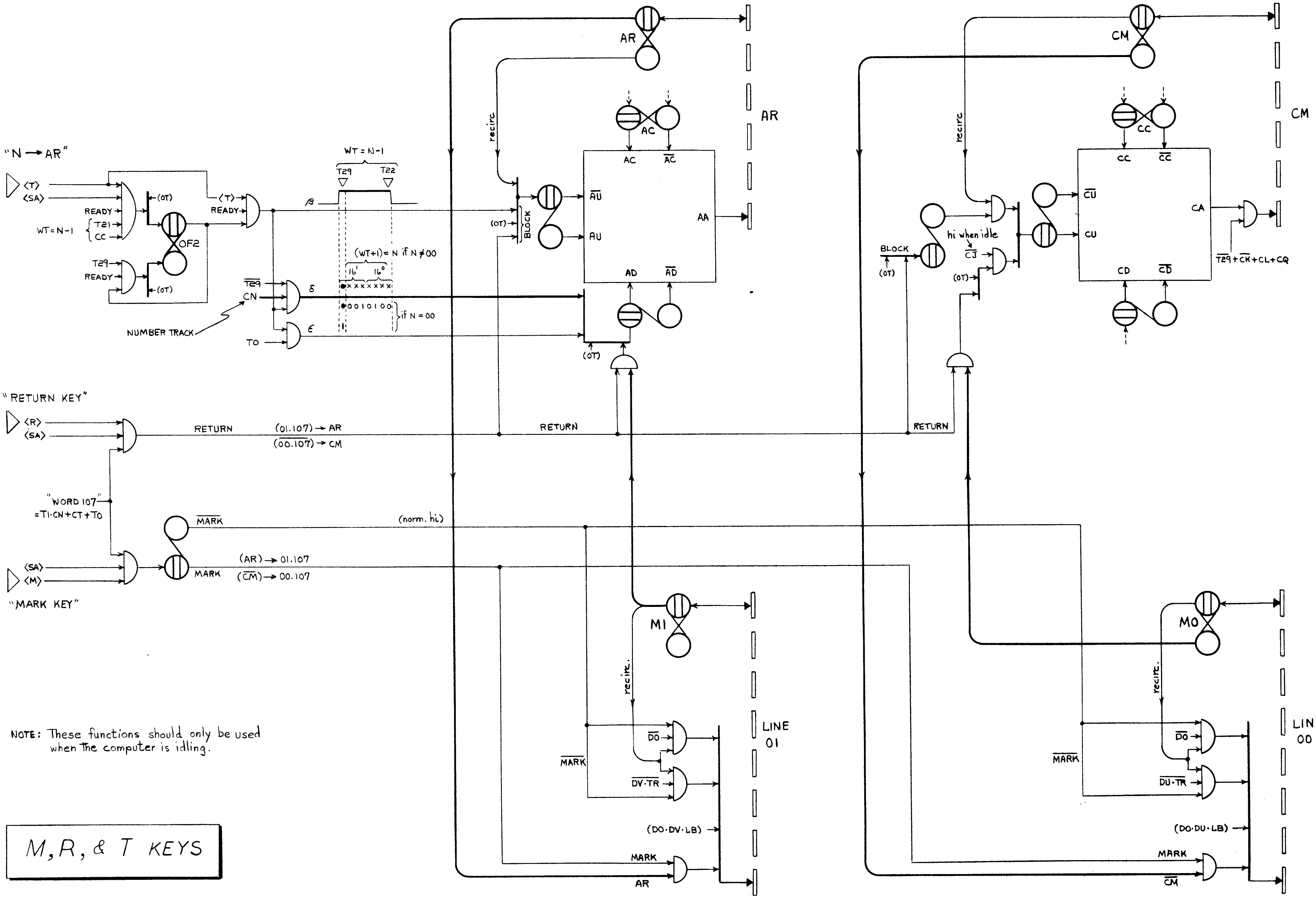
G-15D



POWER SUPPLIES

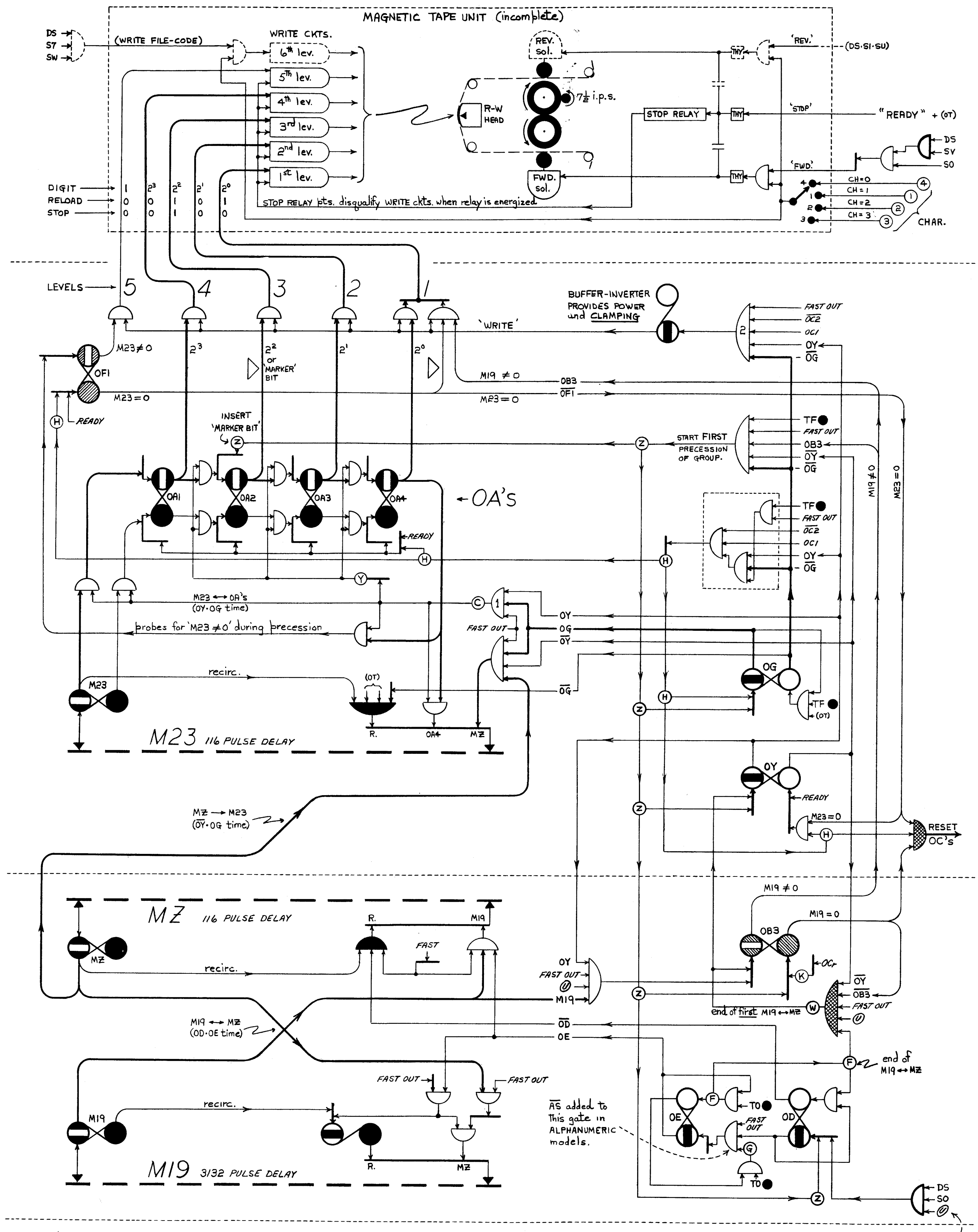
C in Microfarads, R in Ohms, F in Amperes

C1	1500	R4	7500	R20	3385	R109	3300	F5	2
C2	1500	R5	400	R21	5208	R110	3300	F6	2
C3	300	R6	1500	R22	41680	R111	3300	F7	2
C4	270	R7	1000	R28	5000	R112	3300	F8	2
C5	4000	R8	1000	R75	39000	R113	4300	F9	2
C6	4000	R12	4	R100	5000	R114	4300	F10	2
		R13	4	R101	8000			F11	2
R1A	4	R16	31250	R105	4300	F1	8	F12	2
R1B	4	R17	41680	R106	4300	F2	8	F13	2
R2	5	R18	32550	R107	3300	F3	2	F14	2
R3	25	R19	32550	R108	3300	F4	1.5	F-20V	1/16



NOTE: These functions should only be used when the computer is idling.

M, R, & T KEYS



([S<16]→31) (MAG TAPE WRITE)

DS/CV → sets OC's to 0001. This yields: $\overline{OC4}$, $\overline{OC3}$, $\overline{OC2}$, OC1, $\overline{OC4}\cdot\overline{OC3}$, and OC2·OC1

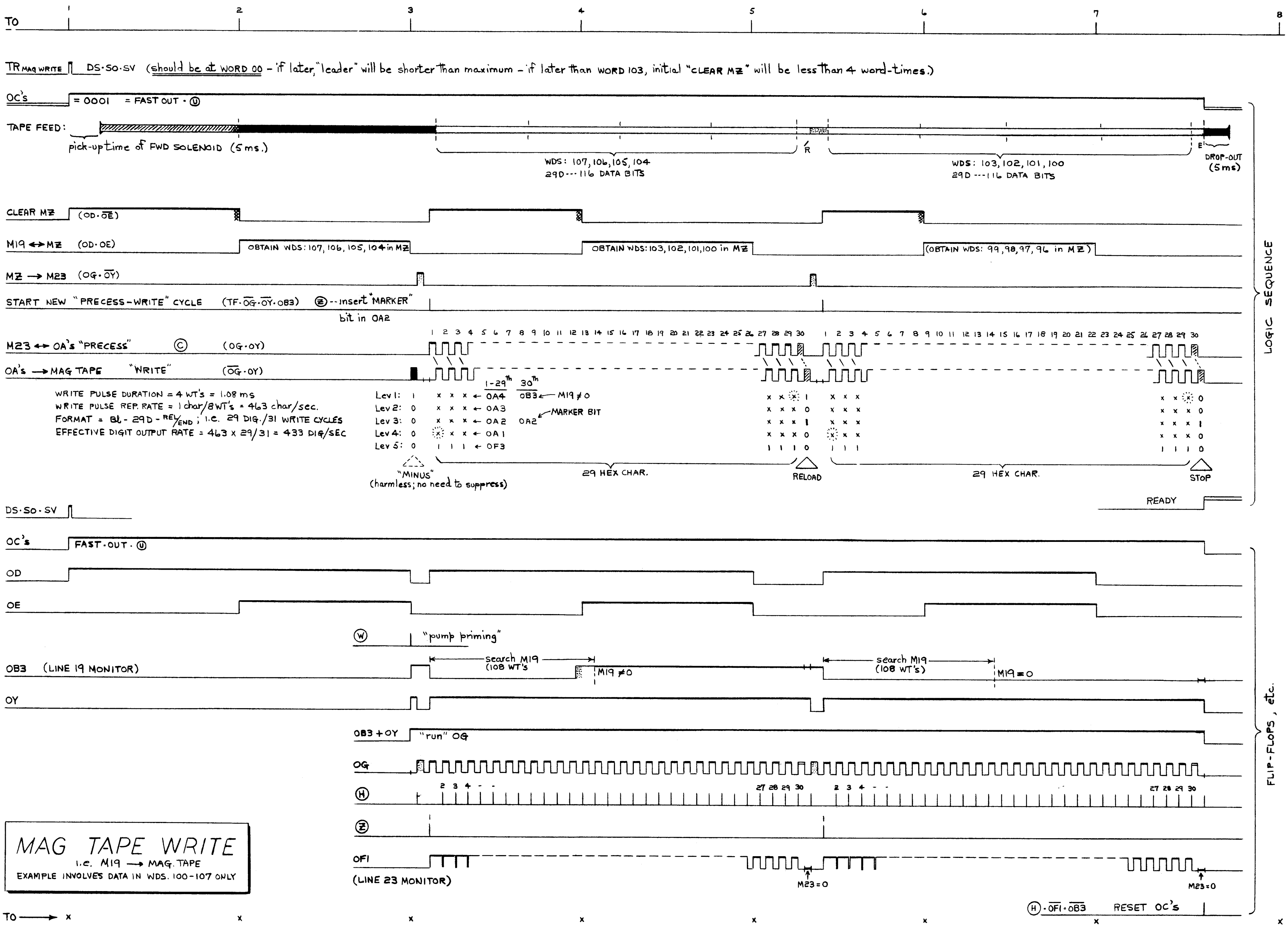
FAST OUT

MAG. TAPE WRITE

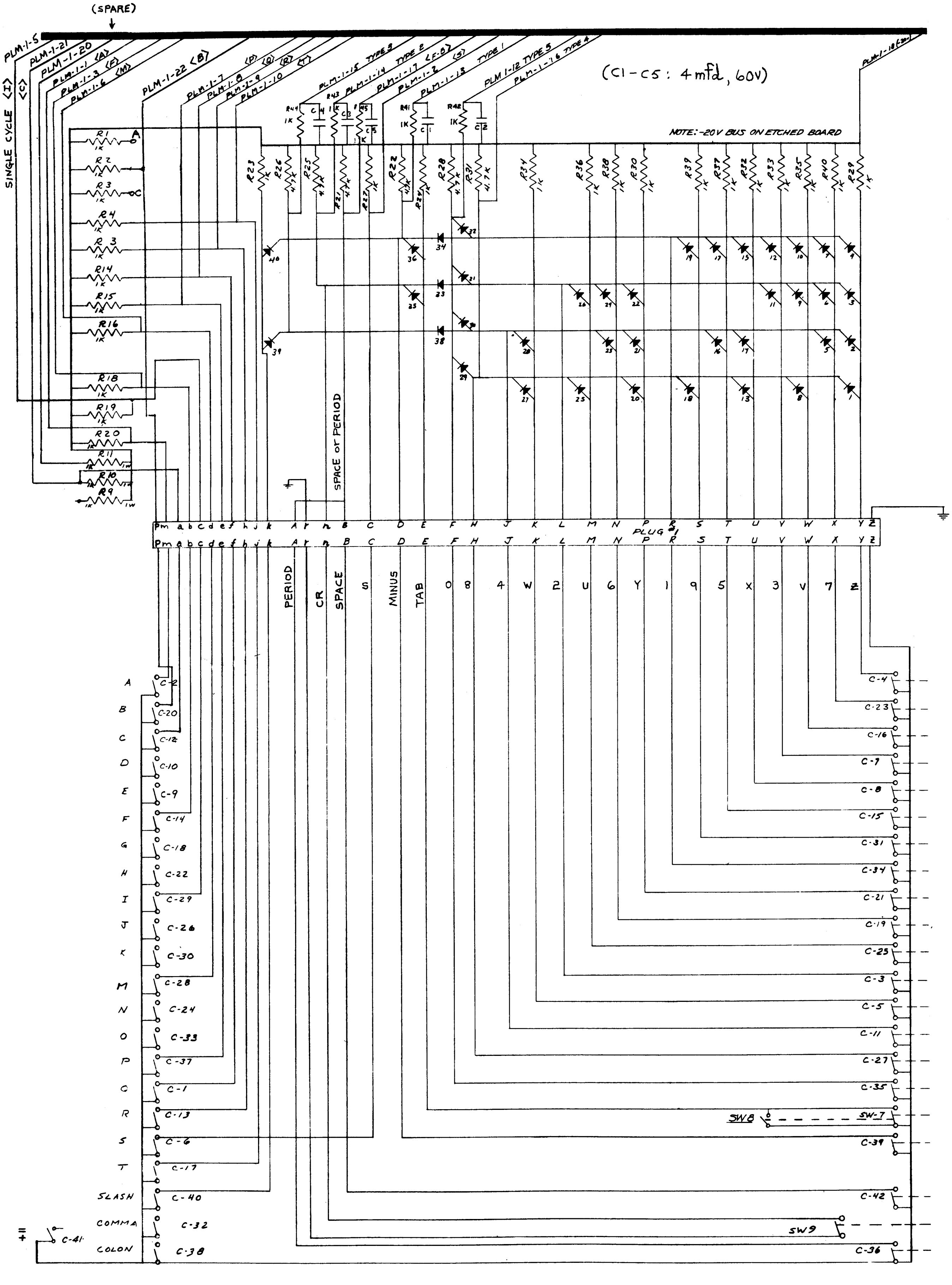
MAG TAPE WRITE

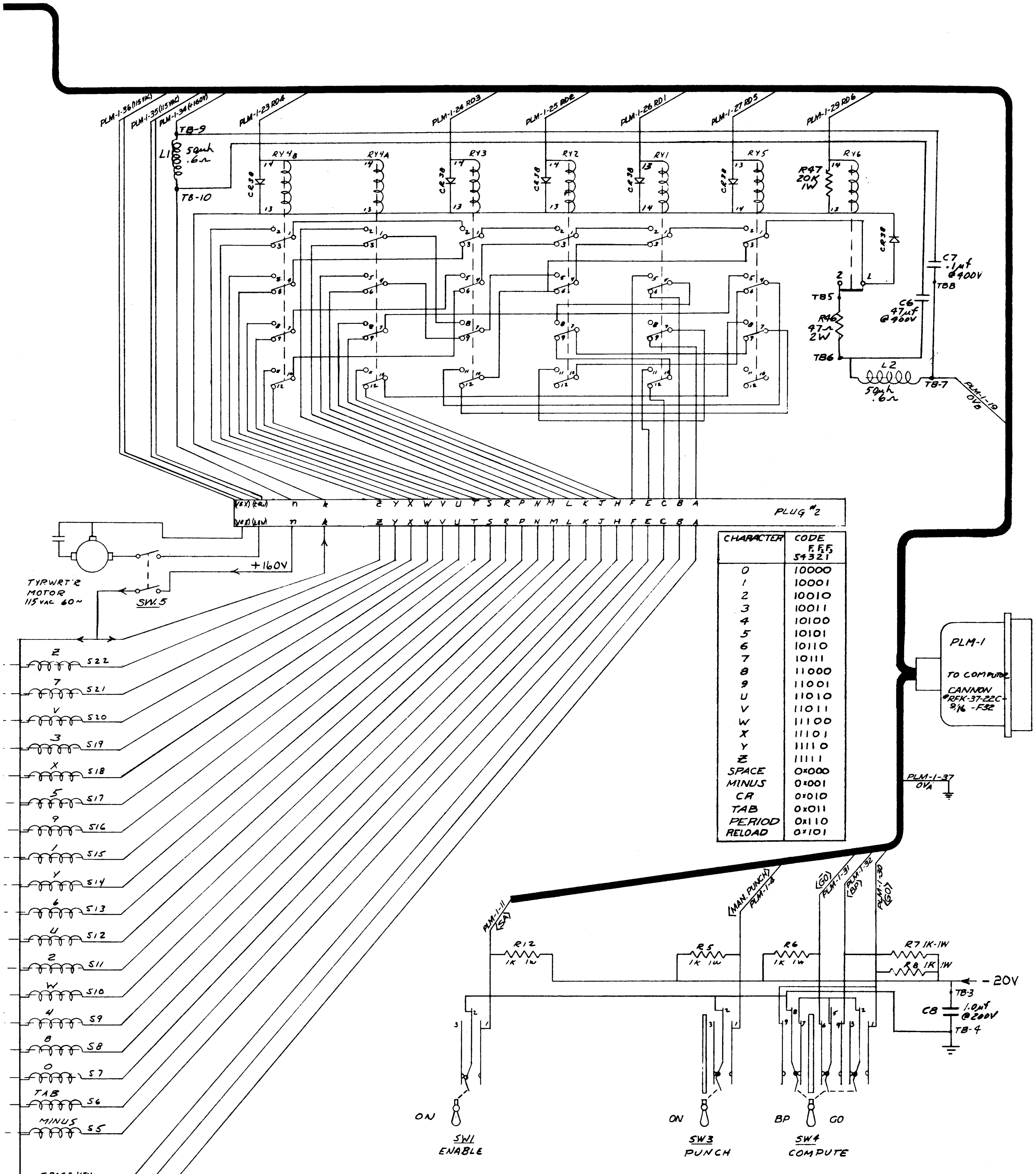
i.e. M19 → MAG. TAPE

EXAMPLE INVOLVES DATA IN WDS. 100-107 ONLY



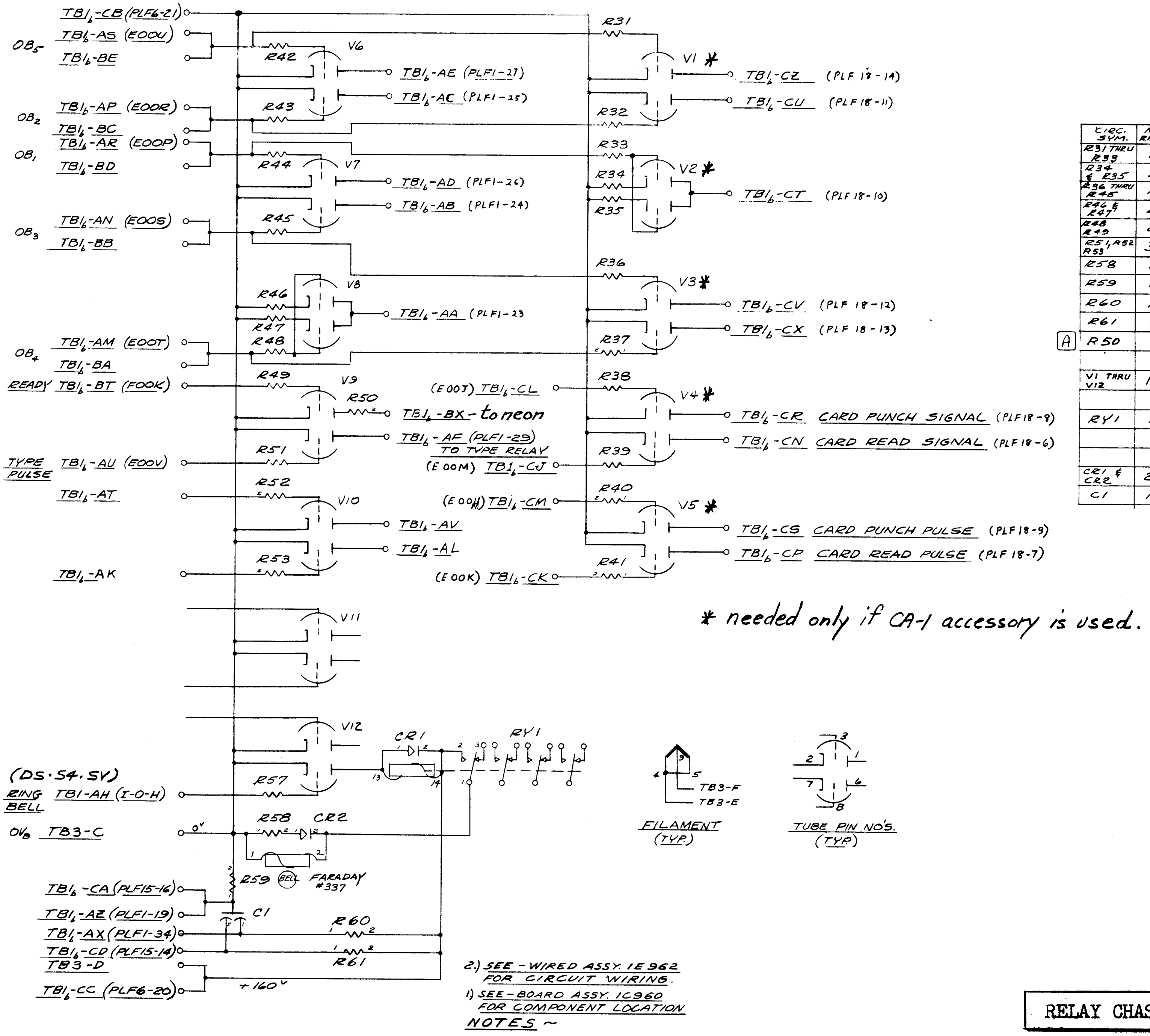
TO
1
2
3
4
5
6
7
8
TR MAG WRITE DS·SO·SV (should be at WORD 00 - if later, "leader" will be shorter than maximum - if later than WORD 103, initial "CLEAR MZ" will be less than 4 word-times.)
OC's = 0001 = FAST OUT · Ⓤ
TAPE FEED:
pick-up time of FWD SOLENOID (5 ms.)
WDS: 107, 106, 105, 104
29D --- 116 DATA BITS
R
WDS: 103, 102, 101, 100
29D --- 116 DATA BITS
E
DROP-OUT (5 ms)
CLEAR MZ (OD·$\overline{OE}$)
M19 ↔ MZ (OD·OE)
OBTAIN WDS: 107, 106, 105, 104 in MZ
OBTAIN WDS: 103, 102, 101, 100 in MZ
(OBTAIN WDS: 99, 98, 97, 96 in MZ)
MZ → M23 (OG·$\overline{OY}$)
START NEW "PRECESS-WRITE" CYCLE (TF·$\overline{OG}$·$\overline{OY}$·OB3) Ⓩ--insert "MARKER" bit in OA2
M23 ↔ OA's "PRECESS" Ⓒ (OG·OY)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
OA's → MAG TAPE "WRITE" ($\overline{OG}$·OY)
WRITE PULSE DURATION = 4 WT's = 1.08 ms
WRITE PULSE REP. RATE = 1 char/8 WT's = 463 char/sec.
FORMAT = BL - 29D - REL/END ; i.e. 29 DIG./31 WRITE CYCLES
EFFECTIVE DIGIT OUTPUT RATE = 463 x 29/31 = 433 DIG/SEC
1-29th
30th
Lev 1: 1 x x x ← OA4 OB3 ← M19 ≠ 0
Lev 2: 0 x x x ← OA3
Lev 3: 0 x x x ← OA2 OA2 ← MARKER BIT
Lev 4: 0 x x x ← OA1
Lev 5: 0 1 1 1 ← OF3
"MINUS" (harmless; no need to suppress)
29 HEX CHAR.
RELOAD
29 HEX CHAR.
STOP
LOGIC SEQUENCE
DS·SO·SV
READY
OC's FAST·OUT·Ⓤ
OD
OE
Ⓦ "pump priming"
OB3 (LINE 19 MONITOR)
search M19 (108 WT's
M19 ≠ 0
search M19 (108 WT's)
M19 = 0
OY
OB3 + OY "run" OG
OG
Ⓗ
2 3 4 - -
27 28 29 30
2 3 4 - -
27 28 29 30
Ⓩ
OF1 (LINE 23 MONITOR)
M23 = 0
M23 = 0
Ⓗ·$\overline{OF1}$·$\overline{OB3}$ RESET OC's
FLIP-FLOPS, etc.
TO → x x x x x x x x


(SPARE)
SINGLE CYCLE <I>
<C>
PLM-1-5
PLM-1-21
PLM-1-20
PLM-1-1 <A>
PLM-1-3 <F>
PLM-1-6 <M>
PLM-1-22 <B>
PLM-1-7
PLM-1-8 <P>
PLM-1-9 <Q>
PLM-1-10 <R>
<T>
PLM-1-15 TYPE 3
PLM-1-14 TYPE 2
PLM-1-17 <F-B>
PLM-1-2 <S>
PLM-1-13 TYPE 1
PLM 1-12 TYPE 5
PLM-1-16 TYPE 4
PLM-1-18
(C1-C5: 4 mfd, 60V)
NOTE: -20V BUS ON ETCHED BOARD
R44 1K
R43 1K
R45 1K
R41 1K
R42 1K
C4
C3
C5
C1
C2
R1 1K
R2 1K
R3 1K
R4 1K
R13 1K
R14 1K
R15 1K
R16 1K
R18 1K
R19 1K
R20 1K
R11 1K 1W
R10 1K 1W
R9 1K 1W
A
B
C
R23
R26 4.7K
R25 4.7K
R21 4.7K
R27 1K
R22 1K
R24 1K
R28 4.7K
R31 4.7K
R34 1K
R36 1K
R38 1K
R30 1K
R39 1K
R37 1K
R32 1K
R33 1K
R35 1K
R40 1K
R29 1K
40
39
36
34
35
33
38
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
SPACE or PERIOD
Pm a b c d e f h j k A r n B C D E F H J K L M N P R S T U V W X Y Z
PLUG #1
PERIOD
CR
SPACE
S
MINUS
TAB
0
8
4
W
2
U
6
Y
1
9
5
X
3
V
7
Z
A C-2
B C-20
C C-12
D C-10
E C-9
F C-14
G C-18
H C-22
I C-29
J C-26
K C-30
M C-28
N C-24
O C-33
P C-37
Q C-1
R C-13
S C-6
T C-17
SLASH C-40
COMMA C-32
COLON C-38
C-41
C-4
C-23
C-16
C-7
C-8
C-15
C-31
C-34
C-21
C-19
C-25
C-3
C-5
C-11
C-27
C-35
SW8
SW-7
C-39
C-42
SW9
C-36



CHARACTER	CODE F_1 F_2 F_3 54321
0	10000
1	10001
2	10010
3	10011
4	10100
5	10101
6	10110
7	10111
8	11000
9	11001
U	11010
V	11011
W	11100
X	11101
Y	11110
Z	11111
SPACE	0x000
MINUS	0x001
CR	0x010
TAB	0x011
PERIOD	0x110
RELOAD	0x101

TYPEWRITER and COUPLER ASSEMBLY



CIRC. SYM.	NO. REQ.	PART NO.	DESCRIPTION
R31 THRU R33	3		RESISTOR 10K ½W ±10%
R34 & R35	2		RESISTOR 100Ω ½W ±10%
R36 THRU R45	10		RESISTOR 10K ½W ±10%
R46 & R47	2		RESISTOR 100Ω ½W ±10%
R48 R49	2		RESISTOR 10K ½W ±10%
R51, R52 R53	3		RESISTOR 10K ½W ±10%
R58	1		RESISTOR 470Ω ½W ±10%
R59	1		RESISTOR 15Ω 5W
R60	1		RESISTOR 50Ω 25W
R61	1		RESISTOR 50Ω 25W
A R50	1		RESISTOR 120K ±10% ½W
V1 THRU V12	12		TUBE-#6350 OR EQUIV
RY1	1	57A5	RELAY
CR1 & CR2	2		RECTIFIER INT. RECT. CR-38
C1	1		CAPACITOR - 75 μf - 75μf. 300V

2.) SEE - WIRED ASSY. 1E962 FOR CIRCUIT WIRING.

1) SEE - BOARD ASSY. 1C960 FOR COMPONENT LOCATION

NOTES ~

3C272

RELAY CHASSIS - LOGIC SECTION

ALPHANUMERIC IN/OUT SYSTEM

The drawings to follow pertain only to ALPHANUMERIC G-15's and the associated TYPEWRITERS and COUPLERS. Drawings 1-64 cover both NUMERIC and ALPHANUMERIC models with the following exceptions:

NUMERIC only		ALPHANUMERIC equivalent
42	-	--
43	-	75-87
45	-	67
48	-	68
52	-	69
55	-	--
62-63	-	75-87
64	-	71

Drawing 66 summarizes the logical differences between the various models of the G-15.

GROUP I: Changes applying to all G15's

1) CG_r add: (<F>·<SA> + TAPE START)·WORD 107
considering item 35, this =
2) CH_s (<F> + <P>)·<SA>·WORD 107 +
<AUTO TAPE START>·WORD 107 } NUMERIC
or
3) CQ_r (<F> + <P>·$\overline{AS}$)·<SA>·WORD 107 +
<AUTO TAPE START>·WORD 107 } ALPHA-NUMERIC

4) MAGNETIC TAPE STOP
old: READY + $[STOP]_{OB}$·Ⓢ
new: READY + $[STOP]_{OB}$·Ⓢ·OC1

5) OB_r old: PHOTO TAPE REV·$[STOP]_{OB}$·Ⓔ
(Ⓚ) new: FAST IN·Ⓔ
(see also item 20 for additional
ALPHANUMERIC change) } for MTA-2

† 6) Ⓠ old: SLOW OUT·OG
new: SLOW OUT·OG·$[\overline{STOP}]_{OB}$
(applies to NUMERIC only; see item
34 for ALPHANUMERIC version)

GROUP II: Changes applying to all ALPHANUMERIC G15's and some modified NUMERIC G15's

7) CQ_s a) old: DS·S5·SW·T1·AR
new: DS·S5·SW·T1·AR·④
b) add: PL18-16 (from CA-2)
8) M20w add: DS·S7·SX·②·M18 } for CA-2

9) "FORWARD" to PHOTO-READER "FORWARD" TRIODE
old: SLOW·IN·OC2·OC1
new: SLOW·IN·OC2·OC1·PERMIT
10) "REVERSE" to PHOTO-READER "REVERSE" TRIODE
old: FAST IN·OC2
new: FAST IN·OC2·PERMIT } for PR-1

GROUP III: Changes applying to ALPHANUMERIC G15's only

11) AS_s a) add: DS·$\overline{CV}$·C1
b) add: <E>·<SA>
* 12) AS_r add: $\overline{ANC-2}$: READY
ANC-2: OC_r
13) AUTO = SLOW IN·AS } (new)

14) HC delete: TYPE·$\overline{OC2}$·$\overline{OF3}$

15) M23w a) apply: $\overline{OC4}+\overline{OC3}+\overline{AS}+\overline{OY}$ (= $\overline{AUTO·OY}$)
to the recirculation gate
b) add: AUTO·OY·$\overline{OG}$·TE·$\overline{CF}$
c) old: IN·OG·OF3·OA1
new: IN·OG·OF3·OA1·$\overline{OH}$

16) OA_r add: AUTO·OH·OS·$\overline{OG}$·TF

17) $OA1_s$ a) add: AUTO·OY·TE·$\overline{CF}$
b) old: M23·Ⓞ
new: M23·Ⓞ·$\overline{OY}$

18) $OA1_r$ a) add: AUTO·OY·T2
b) old: $\overline{M23}$·Ⓞ
new: $\overline{M23}$·Ⓞ·$\overline{OY}$

19) OB_s The "LEVELS ⟶ OB's" probe:
old: IN·OF1
new: IN·OF1·$\overline{OH}$ + IN·OF1·$\overline{OS}$
(= IN·OF1·$\overline{OH·OS}$)

† 20) OB_r old: OC_r
(Ⓚ) new: READY
(see also item 5)

* 21) OC_r a) old: <Ⓢ>·$\overline{OC2}$·$\overline{OC1}$
new: $\overline{ANC-2}$: <Ⓢ>·$\overline{OC2}$·$\overline{OC1}$·$\overline{OH}$
ANC-2: <Ⓢ>·$\overline{OH}$
b) old: $[STOP]_{OB}$·SLOW IN·$\overline{OF1}$·OE
new: $[STOP]_{OB}$·SLOW IN·$\overline{OF2}$·$\overline{OD}$
c) add: AUTO·OH·OF3·Ⓕ
d) add: Ⓜ·OH·$\overline{OA1}$

22) OC_3 add: <E>·<SA>

23) OC_4 add: <E>·<SA>

24) OD_s a) old: $[STOP + REL]_{OB}$·Ⓢ
new: $[STOP + REL]_{OB}$·Ⓢ·$\overline{AS}$
(= $\overline{OB5}$·OB3·$\overline{OB2}$·Ⓢ·$\overline{AS}$)
b) add: AUTO·OG·TF·OA3
c) add: AUTO·OG·TF·M23
d) add: Ⓜ·OH·$\overline{OA1}$

* 25) OD_r add: $\overline{ANC-2}$: $\overline{OC4}$·$\overline{HC}$·TO·AS
ANC-2: $\overline{OC4}$·$\overline{HC}$·TO·OH

* 26) OE_s a) add: SLOW OUT·OH·Ⓖ
b) old: FAST OUT·OD·Ⓖ
new: $\overline{ANC-2}$: FAST OUT·OD·Ⓖ·$\overline{AS}$
ANC-2: FAST OUT·OD·Ⓖ·$\overline{OH}$

27) $OF3_s$ a) add: AUTO·OG·TF·OA3·OH·OS
b) delete: TYPE·OY·Ⓖ

28) OG_s add: AUTO·OH·OS·$\overline{OG}$·TF

29) OH_s a) add: DS·$\overline{CV}$·C1·④·IN
b) add: <E>·<SA>
c) add: TYPE·AS·OY·TO
30) OH_r a) add: READY
b) add: TYPE·$\overline{OY}$·TO } (new)

* 31) OS_s a) add: <Ⓢ>·AUTO
b) add: DS·S2·SV
c) old: T1·CN·M19·Ⓣ
new: $\overline{ANC-2}$: T1·CN·M19·Ⓣ
ANC-2: T1·CN·M19·Ⓣ·$[SIGN]_{OF}$

32) OY_s a) add: DS·$\overline{CV}$·C1·AUTO·TF
b) add: <E>·<SA>·TO
c) add: AUTO·OG·TF·OA3
d) add: AUTO·OG·TF·M23
e) old: $[STOP + REL]_{OB}$·Ⓢ
new: $[STOP + REL]_{OB}$·Ⓢ·$\overline{AS}$
(= $\overline{OB5}$·OB3·$\overline{OB2}$·Ⓢ·$\overline{AS}$)
f) old: SLOW OUT·$\overline{HC}$·Ⓖ
new: SLOW OUT $\overline{HC}$·$\overline{OY}$·$\overline{OH}$·$\overline{OS}$·TO

33) OY_r a) old: TYPE·Ⓕ
new: TYPE·Ⓕ·OY
b) add: TYPE·AS·$\overline{OH}$·<F-B>

34) Ⓠ old: SLOW OUT·OG
new: SLOW OUT·OG·$\overline{AS}$·$[\overline{STOP}]_{OB}$

35) TAPE START
old: <AUTO TAPE START> + <P>·<SA>
new: <AUTO TAPE START> + <P>·<SA>·$\overline{AS}$

36) 5th LEVEL OUTPUT DRIVER
a) old: OB5
new: OB5·$\overline{AS}$
b) add: OB5·$\overline{OY}$·OH

GROUP IV: Additional DRIVER TUBES in LOGIC CHASSIS to accommodate ALPHANUMERIC OUTPUT

		SUBCHASSIS
37) OB1	old: V7A	
	new: V7A,	V13A, V13B
38) OB2	old: V6B	
	new: V6B,	V14A, V14B
39) OB3	old: V7B	
	new: V7B,	V15A, V15B
40) TYPE	new:	V16A
41) AS	new:	V16B

GROUP V: Additional wiring to CONNECTORS to accommodate ALPHANUMERIC INPUT/OUTPUT

* 42) PLF14-32 old: (SPARE)
new: $\overline{ANC-2}$: AS DRIVER output
ANC-2: (SPARE)

* 43) PLF1-18 old: -20V
new: $\overline{ANC-2}$: -20V
ANC-2: AS DRIVER output

44) PLF1-20 old: (SPARE)
new: <E>

45) PLF1-28 old: (SPARE)
new: <REWIND>
(involves addition of RY-C to PHOTO-TAPE READER)

46) PLF1-33 old: (SPARE)
new: TYPE DRIVER output

NOTES

* Differences in logic exist if an ANC-2 is used instead of an ANC-1 or NC-1. The ANC-1 logic is standard, hence is indicated on the drawings.

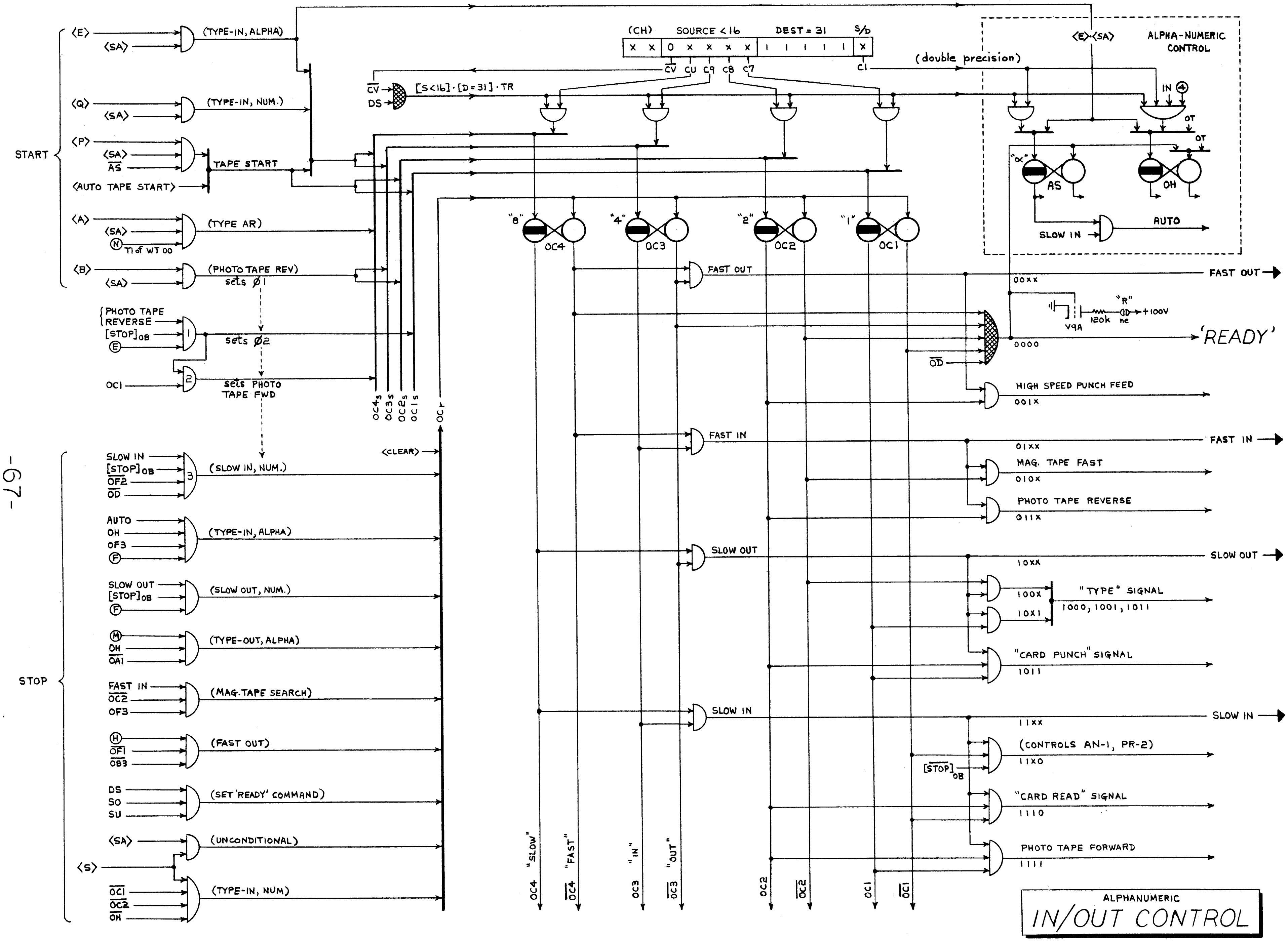
† These changes are anticipated at the time of this publication however, they are not yet applied to the hardware or indicated on any drawings.

<Ⓢ> is the same signal as <S>. The INPUT-OUTPUT WRITER uses the "Ⓢ" key to raise the signal while the original NUMERIC TYPEWRITER uses the "s" key.

GENERAL: This list does not include new equations for G-15 outputs dealing with control of new accessories. Changes to original G-15 logic to accommodate new accessories are listed in GROUP II.

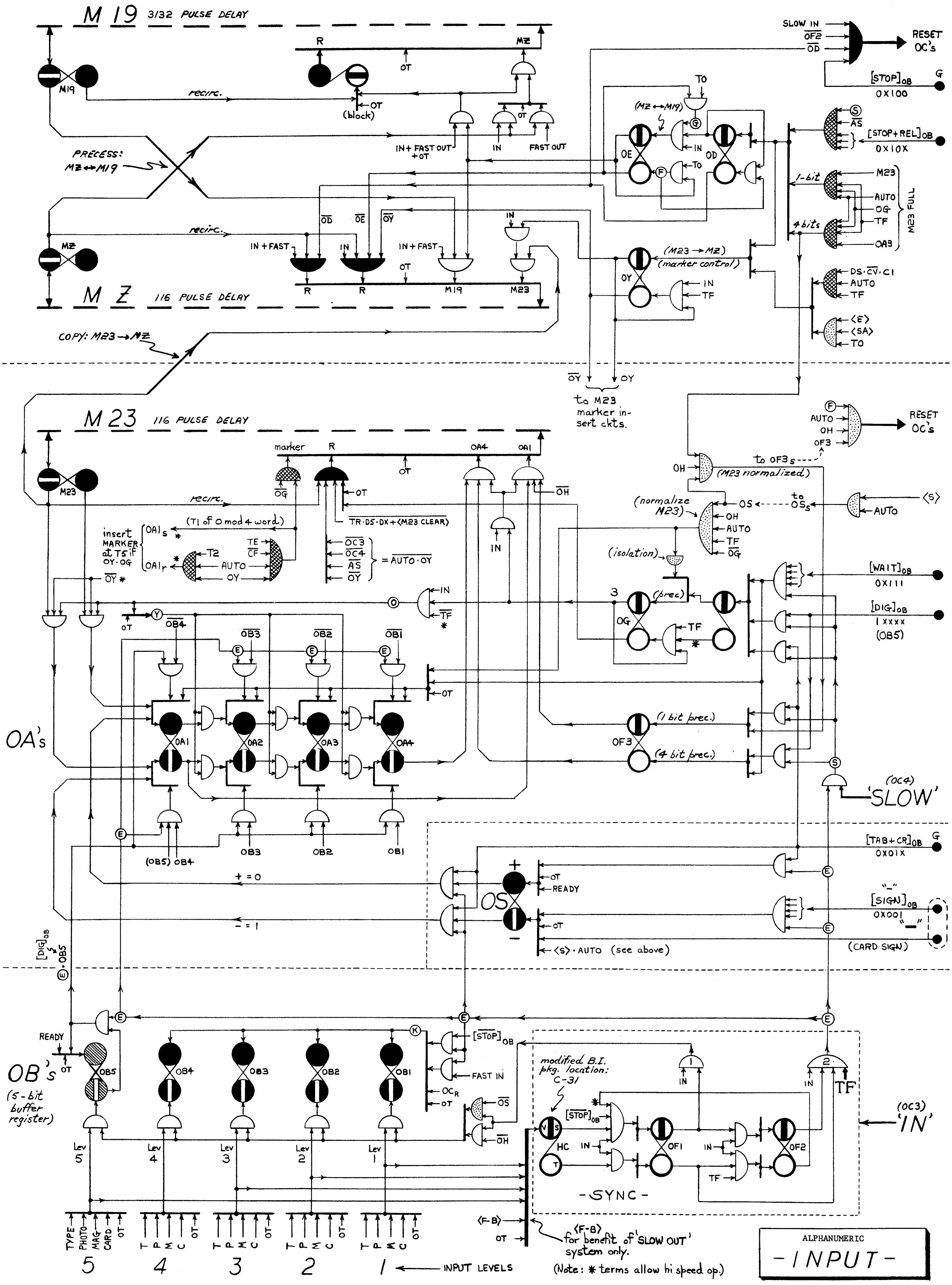
G15 LOGIC CHANGES

resulting from
ECO 970 - 1153



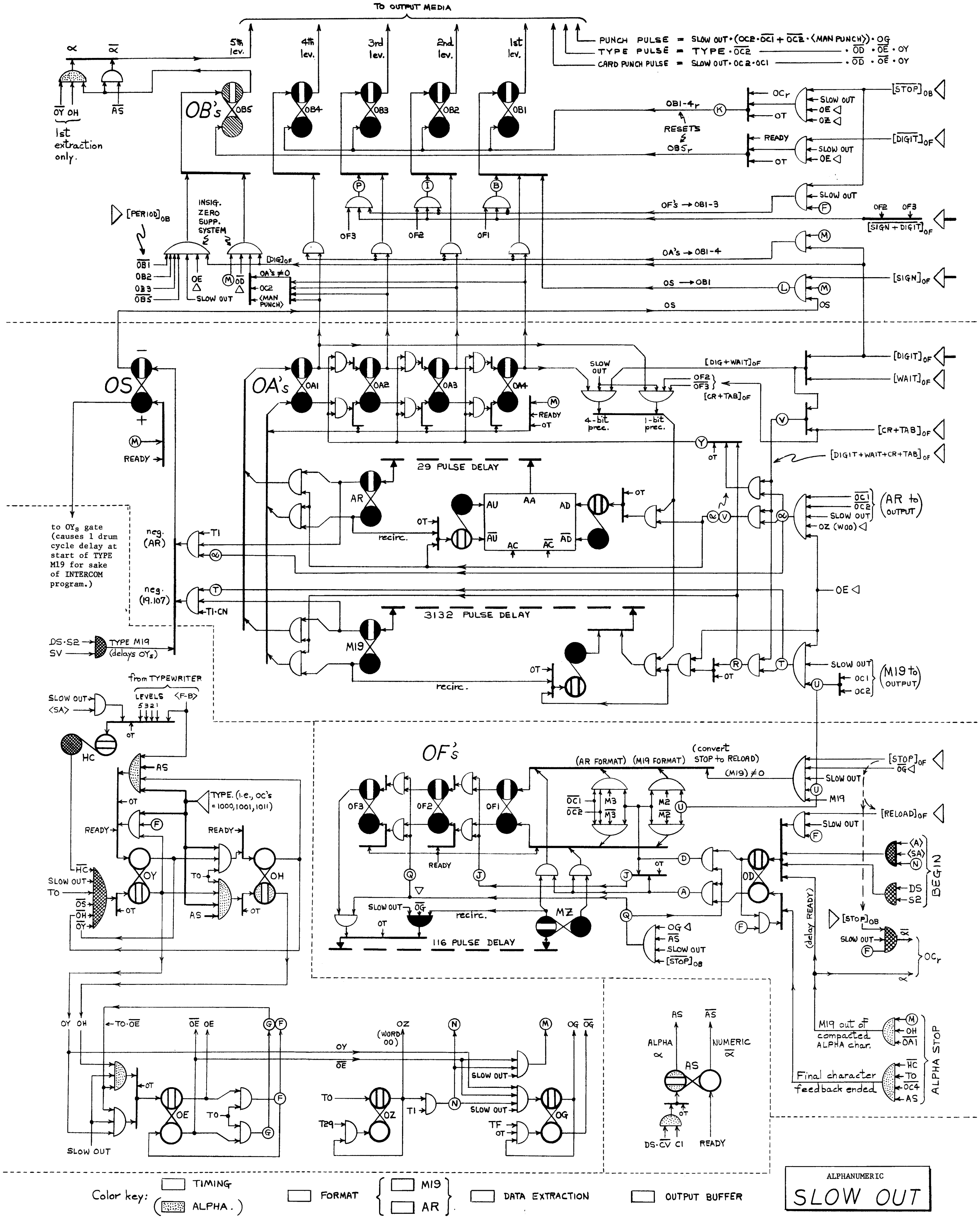
ALPHANUMERIC

IN/OUT CONTROL

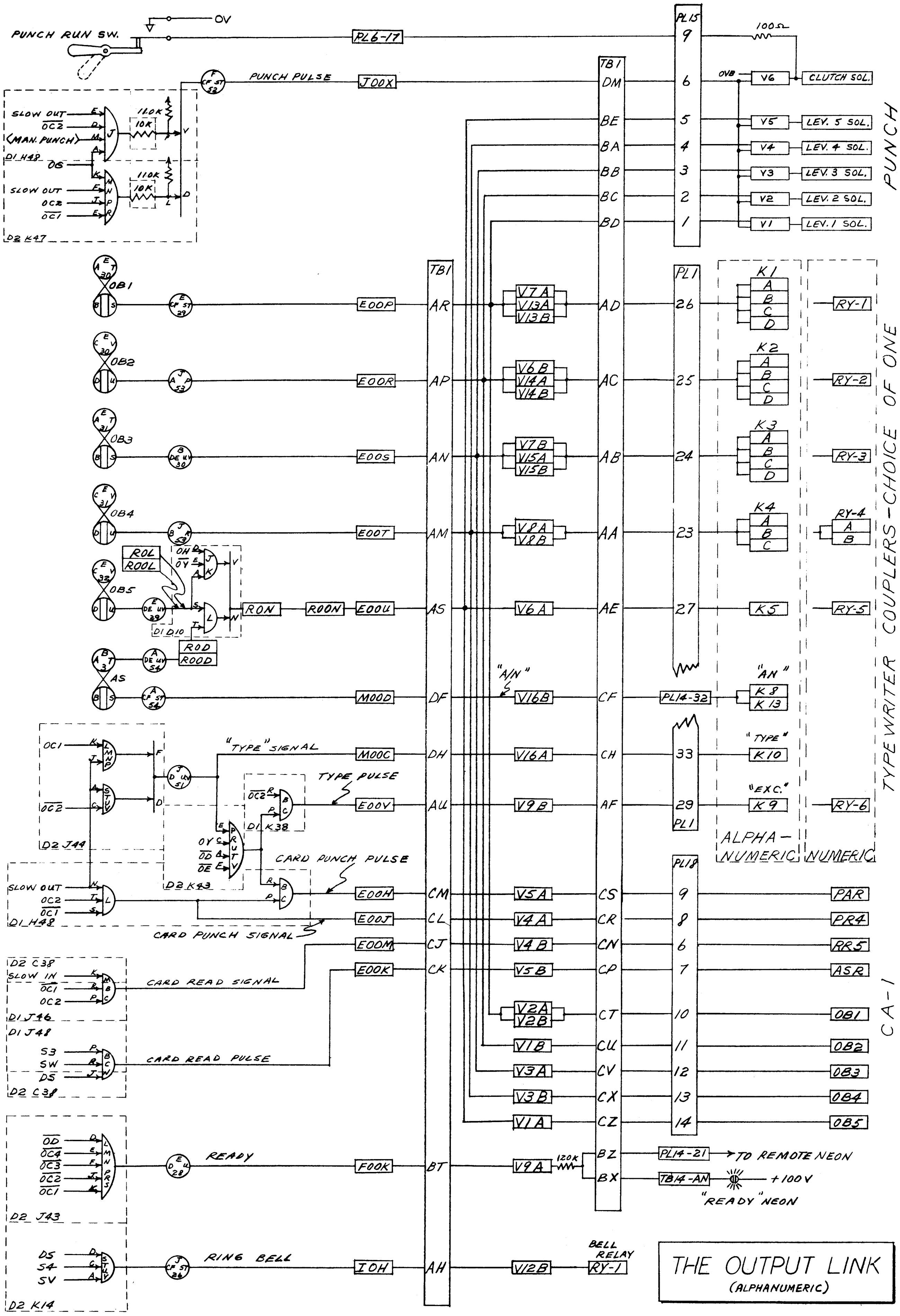


ALPHANUMERIC

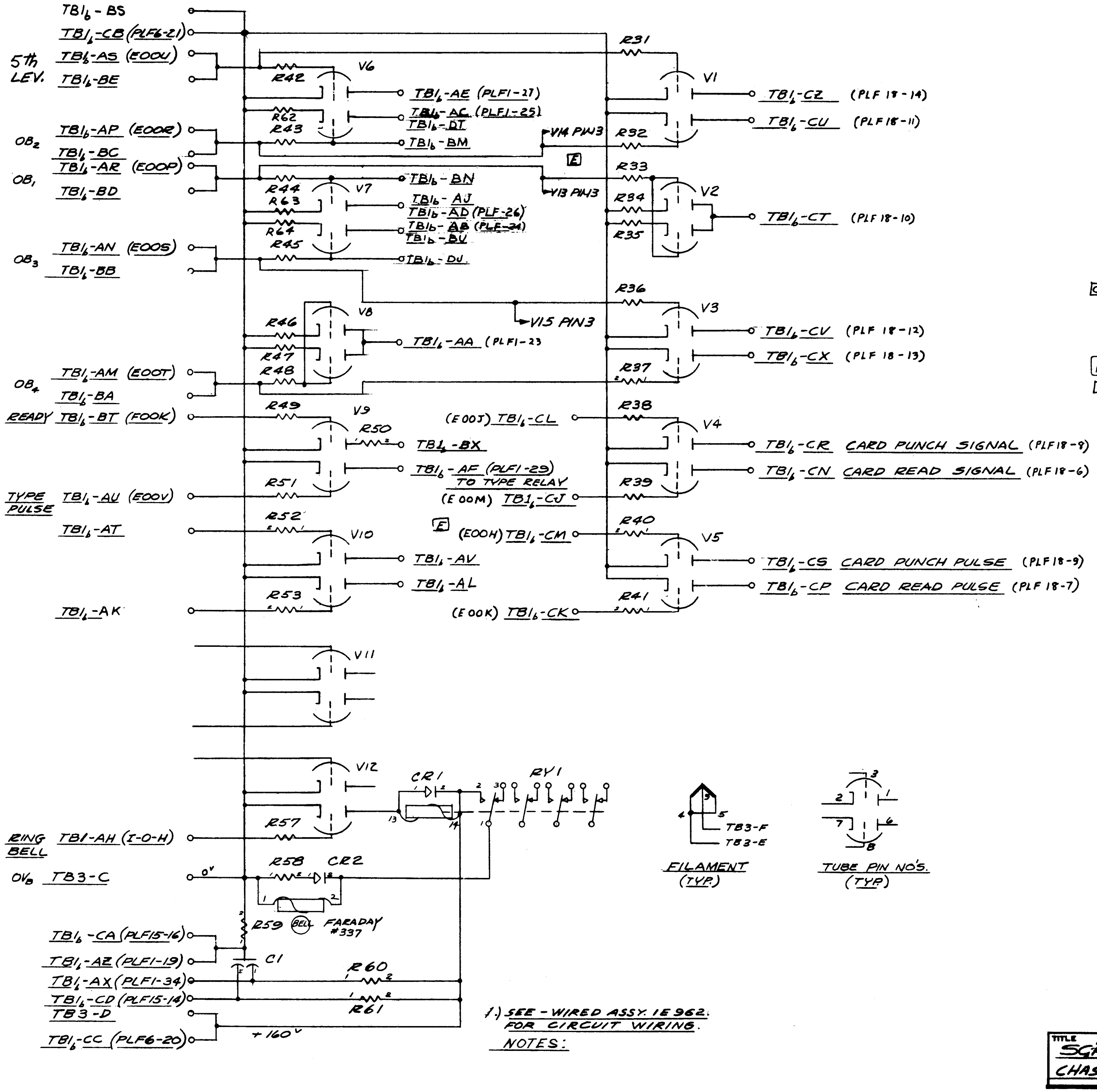
-INPUT-



ALPHANUMERIC
SLOW OUT



THE OUTPUT LINK
(ALPHANUMERIC)



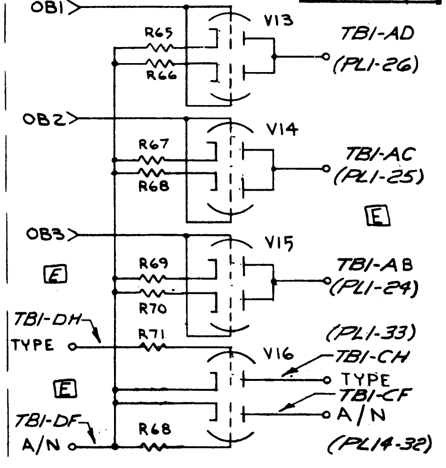
1.) SEE - WIRED ASSY. 1E962. FOR CIRCUIT WIRING.

NOTES:

REVISIONS	
LTR.	DESCRIPTION
A	SEE ECO 542
B	SEE ECO# 804
C	SEE ECO# 905
D	SEE ECO # 1021
E	SEE ECO#1153

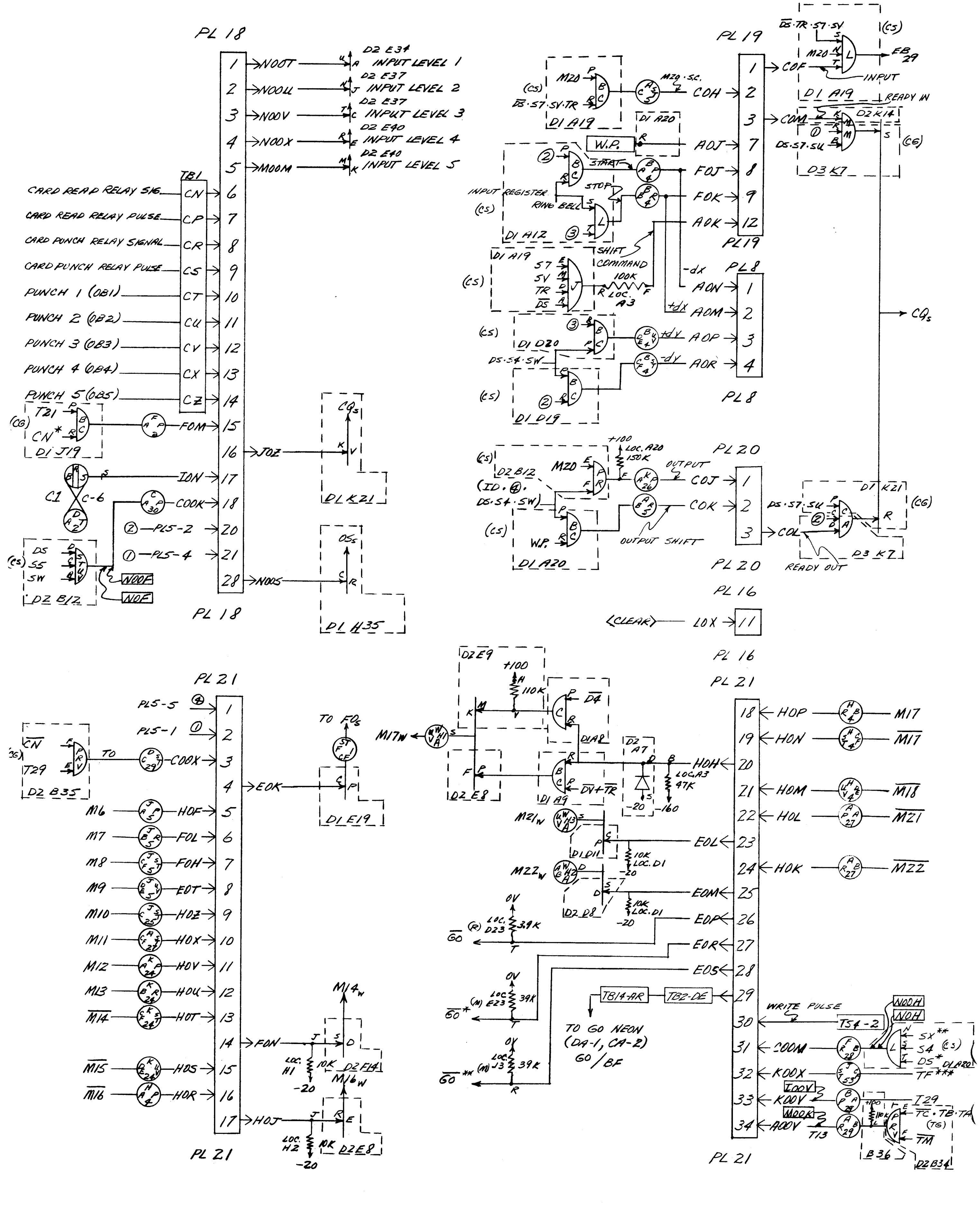
	CIRC. SYM.	NO. REQ.	PART NO.	DESCRIPTION
	R31 THRU R33	3		RESISTOR 10K ½W ±10%
	R34 & R35	2		RESISTOR 100Ω ½W ±10%
	R36 THRU R45	10		RESISTOR 10K ½W ±10%
	R46 & R47	2		RESISTOR 100Ω ½W ±10%
	R48 R49	2		RESISTOR 10K ½W ±10%
	R51, R52 R53	3		RESISTOR 10K ½W ±10%
C	R58	1		RESISTOR 470Ω 1W ±10%
	R59	1		RESISTOR 15Ω 5W
	R60	1		RESISTOR 50Ω 25W
	R61	1		RESISTOR 50Ω 25W
A	R50	1		RESISTOR 120K ±10% ½W
D	R62 THRU R70	9		RESISTOR 100Ω ±10% ½W
D	V1 THRU V18	16		TUBE-#6350 OR EQUIV.
D	R71 & R72	2		RESISTOR 10K ±10% ½W
D	RY1	1	57A62	RELAY
	CR1 & CR2	2		RECTIFIER INT. RECT. CR-38
	C1	1		CAPACITOR - 75 μf - 75μf. 300V

	CHG. LTR.
3C272	E

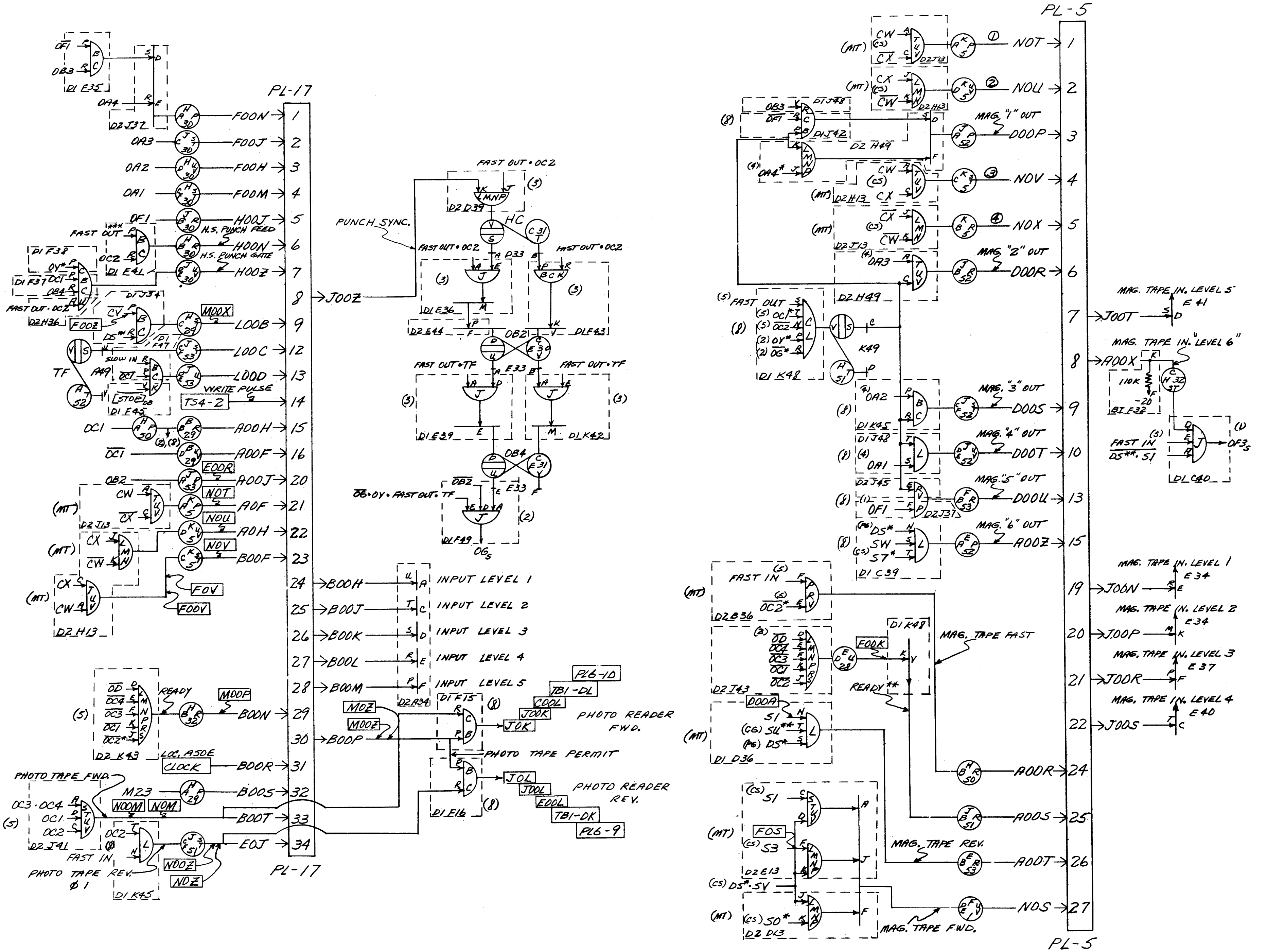


A/N TYPEWRITER MODIFICATION

TITLE SCHEMATIC - REL CHASSIS	PART NO. 3C272	CHG. LTR. E



ACCESSORY CONTROL
1



ACCESSORY CONTROL
2

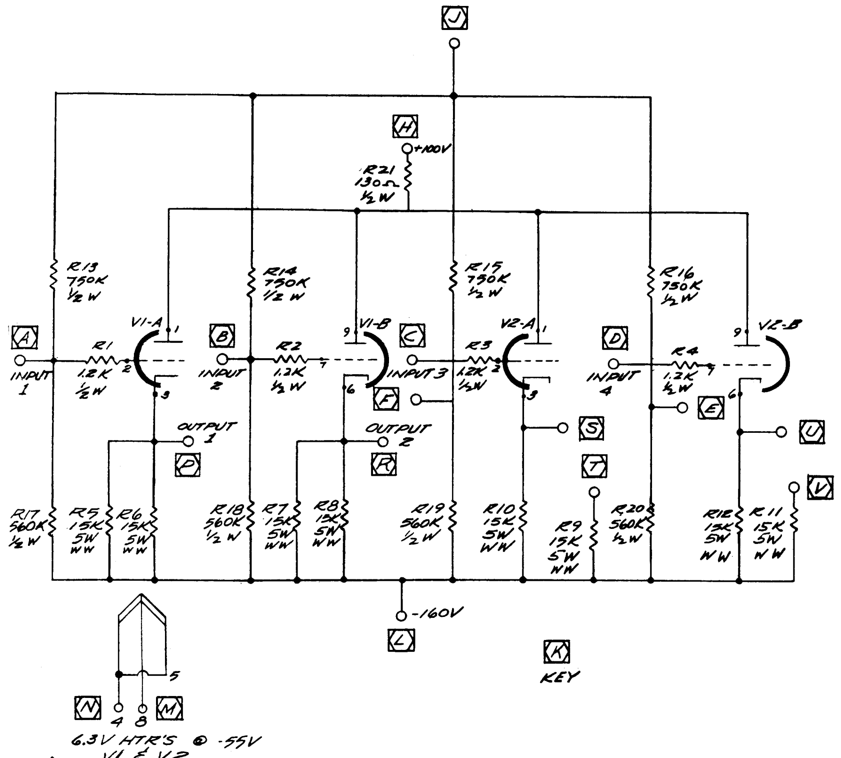
ITEM NO.	NO. REQ.	PART NO.	DESCRIPTION
1	1	39C9-002A	HANDLE-PLUG-IN PACKAGE
2	1	7B2-005F	PANEL-PLUG IN PKG TUBE
3	1	1B2099	BOARD ASSY-CATH FOL 3
4	2	62B2 A	SOCKET - 9 PIN TUBE
5	6	15C1-004B	RIVET - SEMI-TUBULAR
6	4	14C1-019B	EYELETS

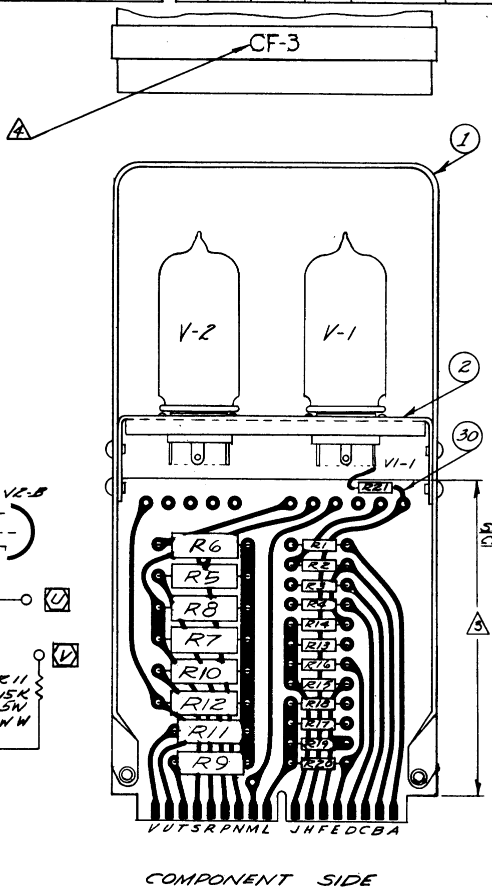
ZONE	ITEM NO.	CIRC. SYM.	NO. REQ.	PART NO.	DESCRIPTION
	7	R1	1	80C10-122B	RESISTOR-½W 5 PRCT
	8	R2	1	80C10-122B	
	9	R3	1	80C10-122B	
	10	R4	1	80C10-122B	RESISTOR-½W 5 PRCT
	11	R5	1	81A46 A	RESISTOR - 5W
	12	R6	1	81A46 A	
	13	R7	1	81A46 A	
	14	R8	1	81A46 A	
	15	R9	1	81A46A	
	16.	R10	1	81A46A	
	17	R11	1	81A46A	
	18	R12	1	81A46 A	RESISTOR - 5W

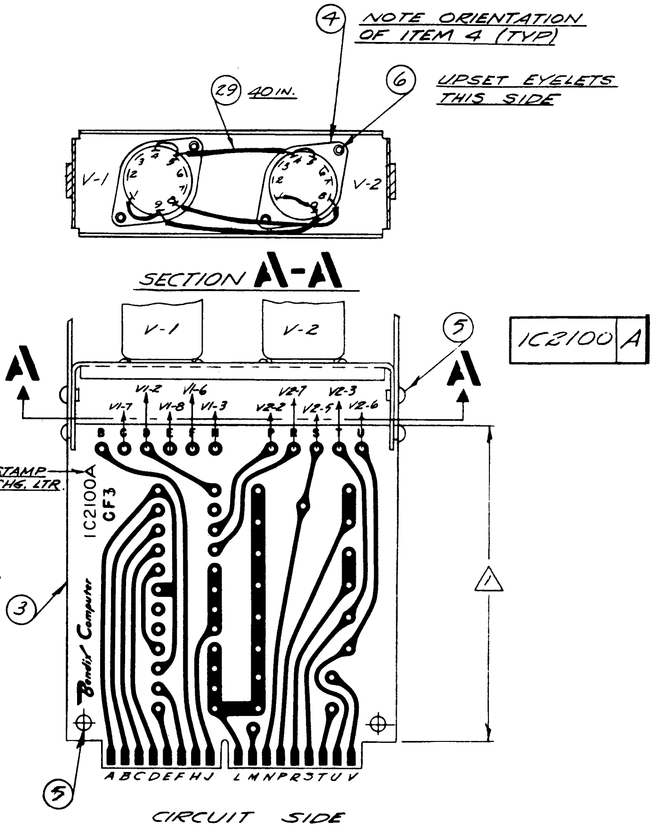
ZONE	ITEM NO.	CIRC. SYM.	NO. REQ.	PART NO.	DESCRIPTION
	19	R13	1	80C10-754B	RESISTOR-½W 5 PRCT
	20	R14	1	80C10-754B	
	21	R15	1	80C10-754B	
	22	R16	1	80C10-754B	RESISTOR-½W 5 PRCT
	23	R17	1	80C10-564B	RESISTOR-½W 5 PRCT
	24	R18	1	80C10-564B	
	25	R19	1	80C10-564B	
	26	R20	1	80C10-564B	RESISTOR-½W 5 PRCT
	27	R21	1	80C10-131B	RESISTOR-½W 5 PRCT
	28	V1 V2	2	93A16 A	TUBE - 5687
	29			67C69-024C	WIRE-ELECT SOLID INSUL
	30		AS REQ.	66C2-003C	TUBING-CLEAR VINYL

REVISIONS

LTR.	DESCRIPTION	DATE	APPR.
A	PRODUCTION RELEASE		







4△ MARK WITH .18 HIGH BLACK LETTERS CENTERED AS SHOWN.

3△ DIP SOLDER SHOWN SIDE WITHIN DIM. NOTED BEFORE COMPONENT INSTALLATION

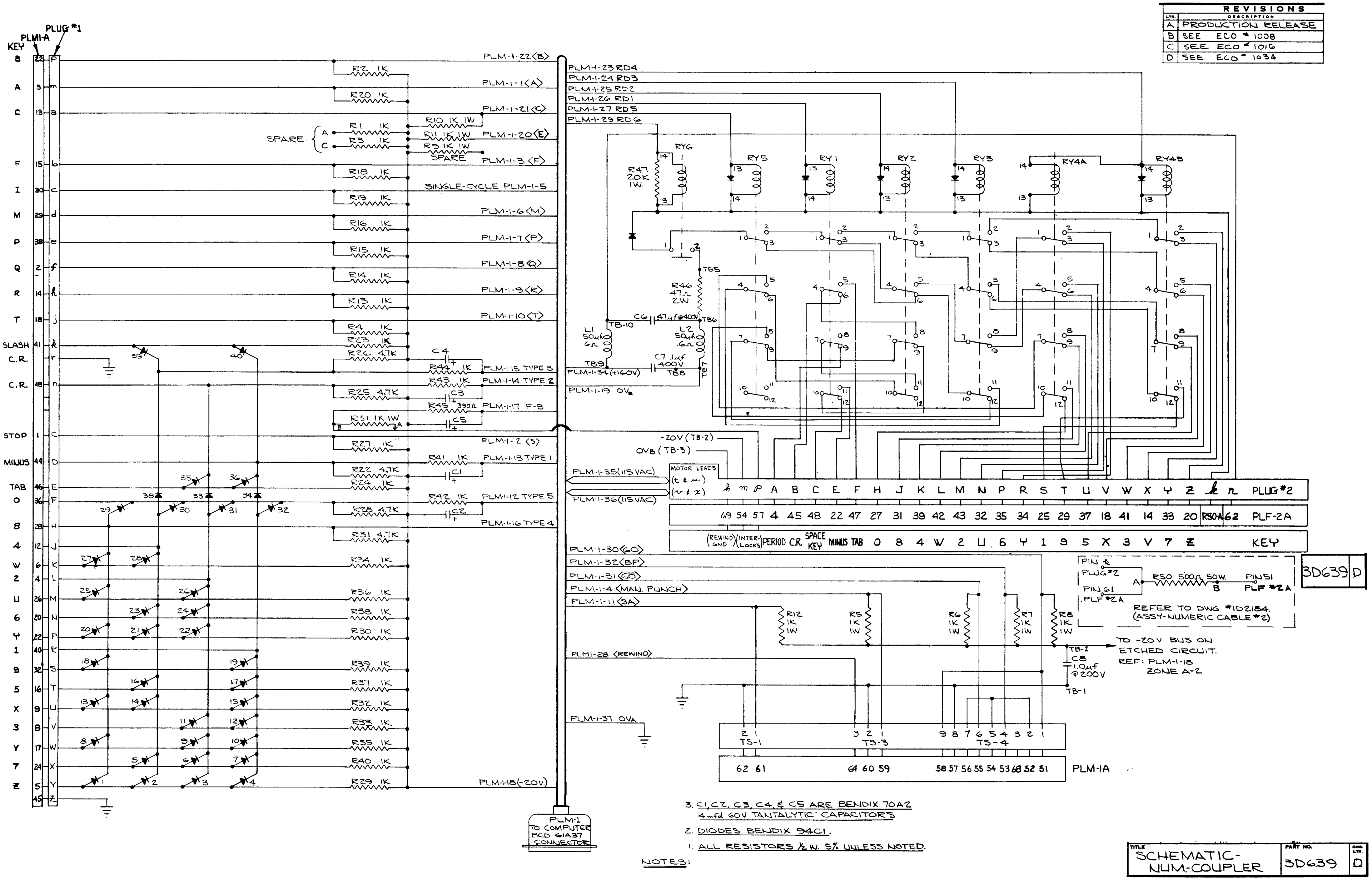
2.) □ INDICATES PIN LETTER ON CONNECTOR.

1△ DIP SOLDER SHOWN SIDE WITHIN DIM. NOTED AFTER COMPONENT INSTALLATION

NOTES:

TITLE	PART NO.	CHG. LTR.
FINAL ASSY - CATH FOL 3	1C2100	A

REVISIONS	
LTR.	DESCRIPTION
A	PRODUCTION RELEASE
B	SEE ECO # 1008
C	SEE ECO # 1016
D	SEE ECO # 1034



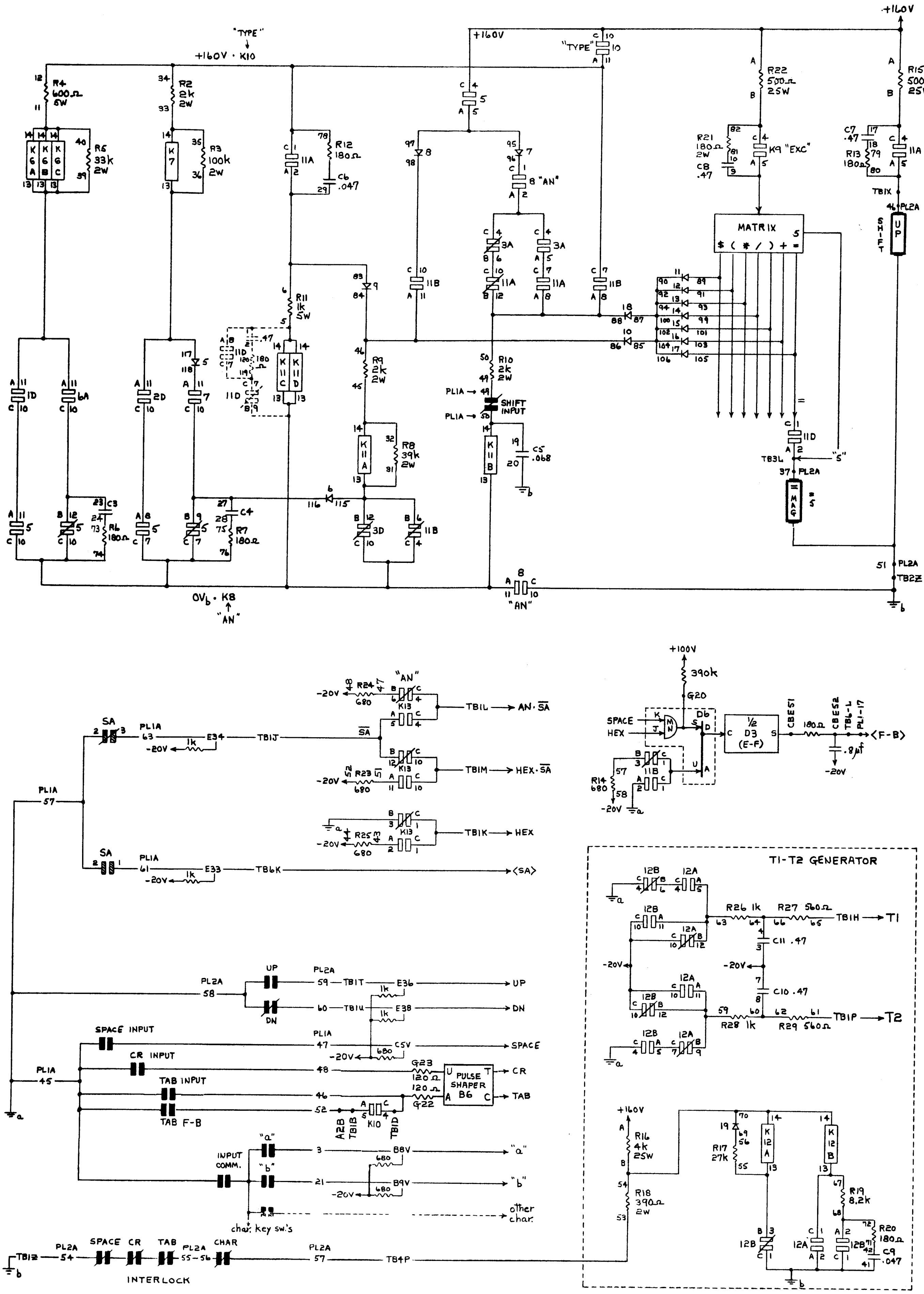
3. C1, C2, C3, C4, & C5 ARE BENDIX 70A2 4mfd 60V TANTALYTIC CAPACITORS

2. DIODES BENDIX 94C1.

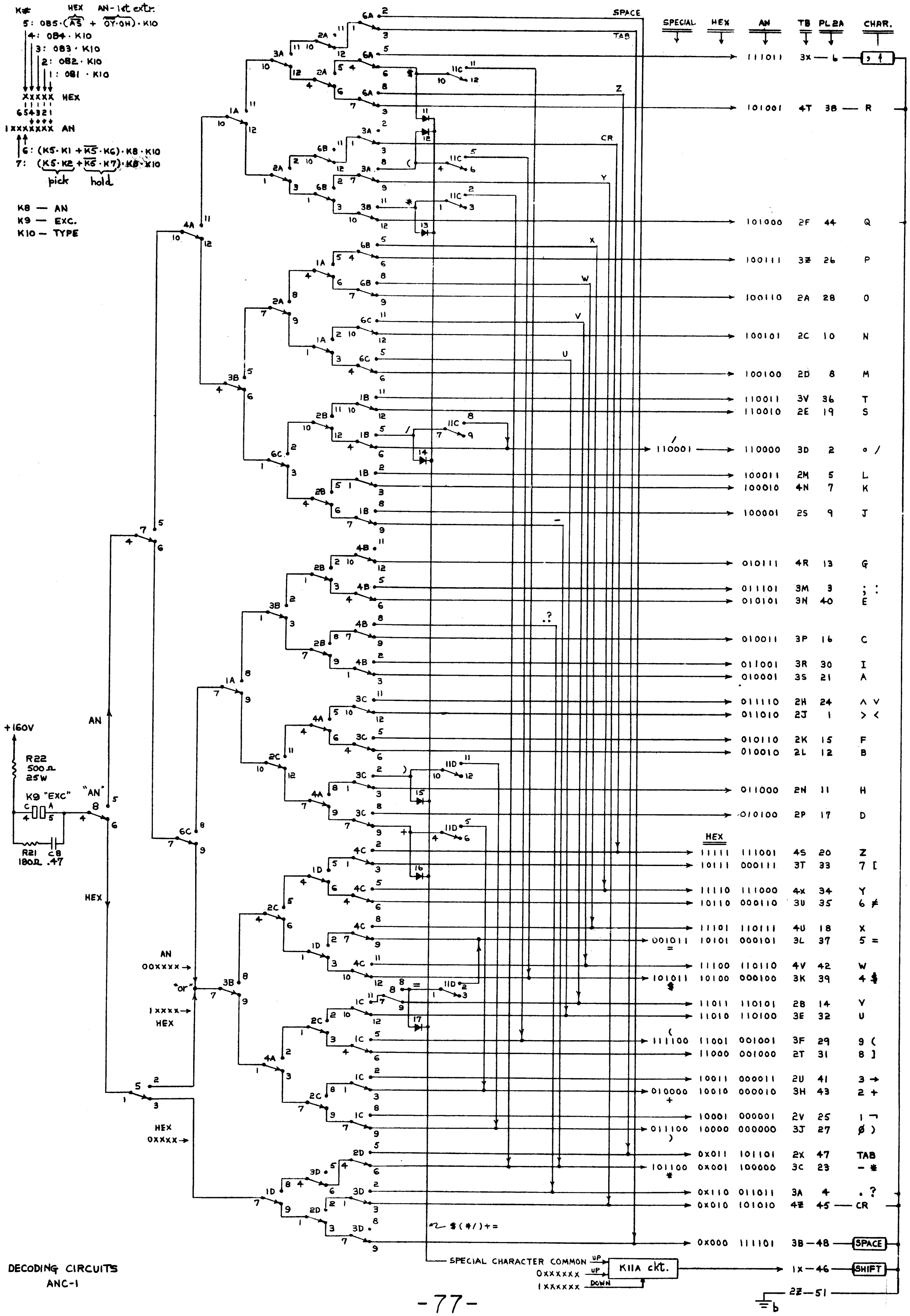
1. ALL RESISTORS ½W. 5% UNLESS NOTED.

NOTES:

TITLE	PART NO.	CHG. LTR.
SCHEMATIC-NUM-COUPLER	3D639	D



CONTROL CIRCUTS, ANC-1



K#

$5:\ \text{OB5}\cdot(\overbrace{\overline{\text{AS}}}^{\text{HEX}} + \overbrace{\overline{\text{OY}}\cdot\text{OH})}^{\text{AN-1st extr.}}\cdot\text{K10}$

$4:\ \text{OB4}\cdot\text{K10}$

$3:\ \text{OB3}\cdot\text{K10}$

$2:\ \text{OB2}\cdot\text{K10}$

$1:\ \text{OB1}\cdot\text{K10}$

XXXXX HEX

654321

1XXXXXXX AN

$6:\ (\text{K5}\cdot\text{K1} + \overline{\text{K5}}\cdot\text{K6})\cdot\text{K8}\cdot\text{K10}$

$7:\ (\underbrace{\text{K5}\cdot\text{K2}}_{\text{pick}} + \underbrace{\overline{\text{K6}}\cdot\text{K7}}_{\text{hold}})\cdot\text{K8}\cdot\text{K10}$

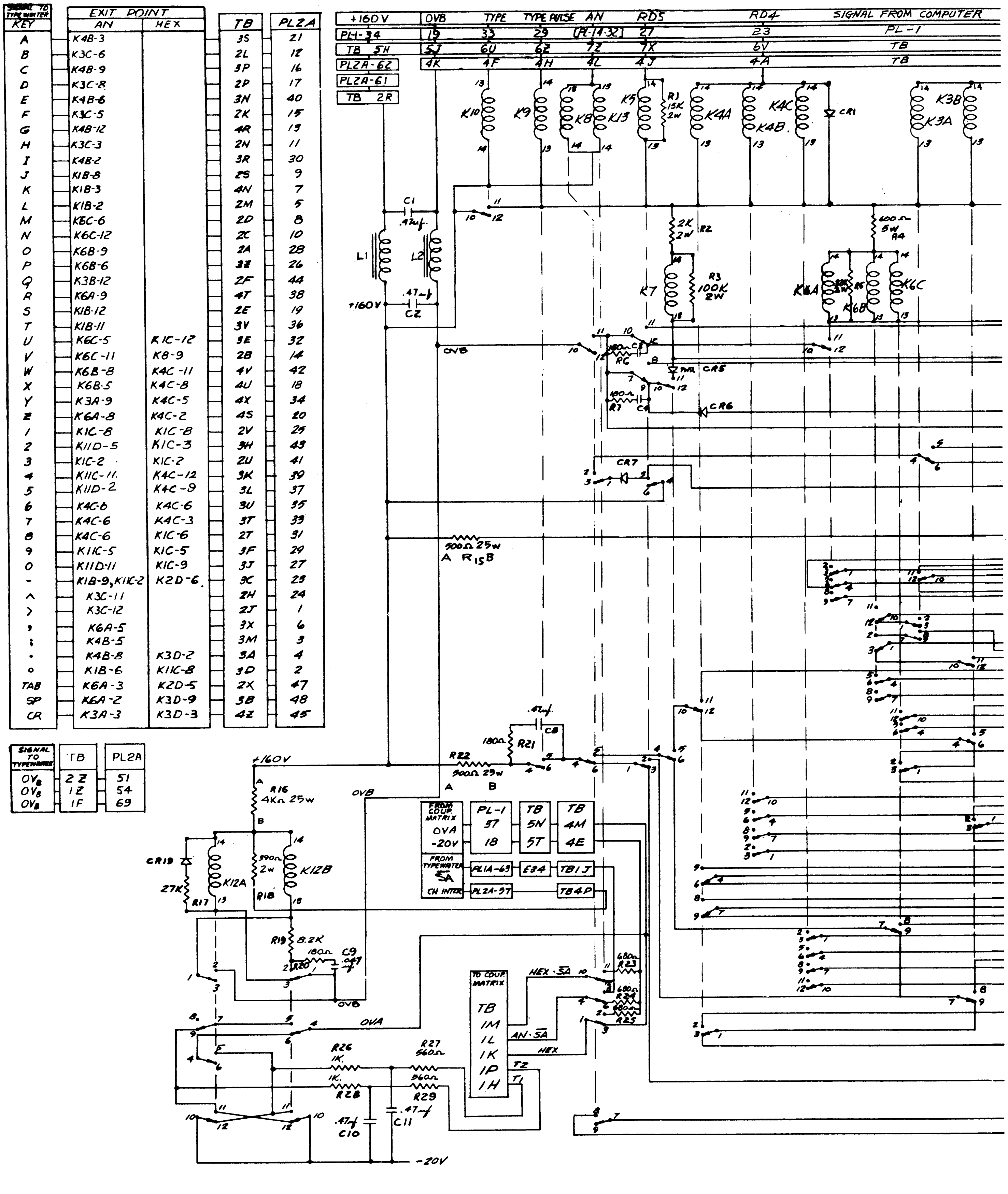
K8 — AN

K9 — EXC.

K10 — TYPE

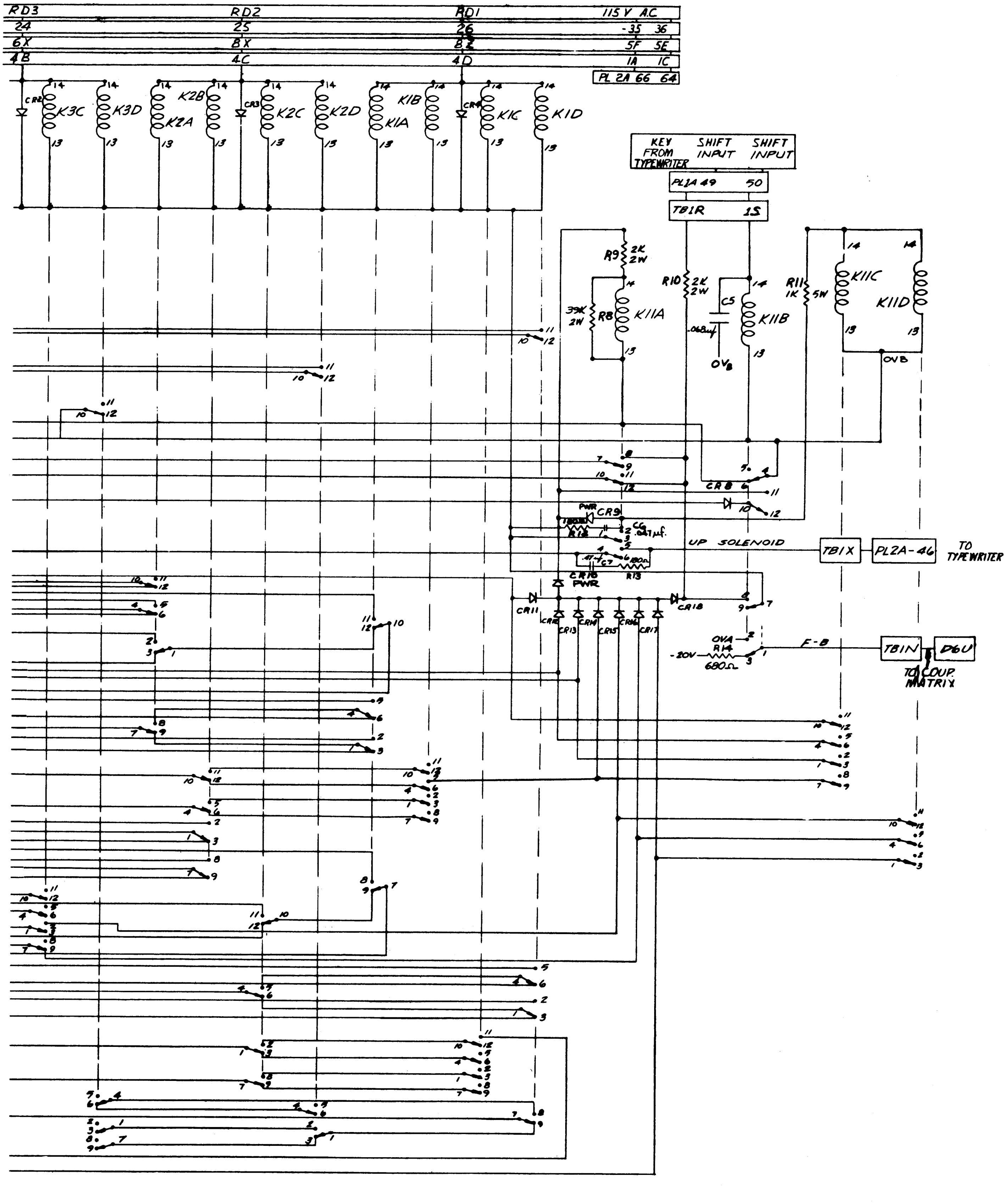
SPECIAL	HEX	AN	TB	PL2A	CHAR.
		111011	3x	6	, ↑
		101001	4T	38	R
		101000	2F	44	Q
		100111	3Z	26	P
		100110	2A	28	O
		100101	2C	10	N
		100100	2D	8	M
		110011	3V	36	T
		110010	2E	19	S
110001 /		110000	3D	2	o /
		100011	2M	5	L
		100010	4N	7	K
		100001	2S	9	J
		010111	4R	13	G
		011101	3M	3	; :
		010101	3N	40	E
		010011	3P	16	C
		011001	3R	30	I
		010001	3S	21	A
		011110	2H	24	∧ ∨
		011010	2J	1	> <
		010110	2K	15	F
		010010	2L	12	B
		011000	2N	11	H
		010100	2P	17	D
	HEX				
	11111	111001	4S	20	Z
	10111	000111	3T	33	7 [
	11110	111000	4X	34	Y
	10110	000110	3U	35	6 ≠
	11101	110111	4U	18	X
001011 =	10101	000101	3L	37	5 =
	11100	110110	4V	42	W
101011 $	10100	000100	3K	39	4 $
	11011	110101	2B	14	V
	11010	110100	3E	32	U
(111100	11001	001001	3F	29	9 (
	11000	001000	2T	31	8]
	10011	000011	2U	41	3 →
010000 +	10010	000010	3H	43	2 +
	10001	000001	2V	25	1 ¬
011100)	10000	000000	3J	27	ø)
	0X011	101101	2X	47	TAB
101100 *	0X001	100000	3C	23	- *
	0X110	011011	3A	4	. ?
	0X010	101010	4Z	45	CR
	0X000	111101	3B	48	SPACE
			1X	46	SHIFT
			2Z	51	

DECODING CIRCUITS
ANC-1



SIGNAL TO TYPEWRITER KEY	EXIT POINT AN	EXIT POINT HEX	TB	PL2A
A	K4B-3		3S	21
B	K3C-6		2L	12
C	K4B-9		3P	16
D	K3C-8		2P	17
E	K4B-6		3N	40
F	K3C-5		2K	15
G	K4B-12		4R	13
H	K3C-3		2N	11
I	K4B-2		3R	30
J	K1B-8		2S	9
K	K1B-3		4N	7
L	K1B-2		2M	5
M	K6C-6		2D	8
N	K6C-12		2C	10
O	K6B-9		2A	28
P	K6B-6		3Z	26
Q	K3B-12		2F	44
R	K6A-9		4T	38
S	K1B-12		2E	19
T	K1B-11		3V	36
U	K6C-5	K1C-12	3E	32
V	K6C-11	K8-9	2B	14
W	K6B-8	K4C-11	4V	42
X	K6B-5	K4C-8	4U	18
Y	K3A-9	K4C-5	4X	34
Z	K6A-8	K4C-2	4S	20
/	K1C-8	K1C-8	2V	25
2	K11D-5	K1C-3	3H	43
3	K1C-2	K1C-2	2U	41
4	K11C-11	K4C-12	3K	39
5	K11D-2	K4C-9	3L	37
6	K4C-6	K4C-6	3U	35
7	K4C-6	K4C-3	3T	33
8	K4C-6	K1C-6	2T	31
9	K11C-5	K1C-5	3F	29
0	K11D-11	K1C-9	3J	27
-	K1B-9, K11C-2	K2D-6	3C	23
^	K3C-11		2H	24
>	K3C-12		2J	1
,	K6A-5		3X	6
;	K4B-5		3M	3
•	K4B-8	K3D-2	3A	4
°	K1B-6	K11C-8	3D	2
TAB	K6A-3	K2D-5	2X	47
SP	K6A-2	K3D-9	3B	48
CR	K3A-3	K3D-3	4Z	45

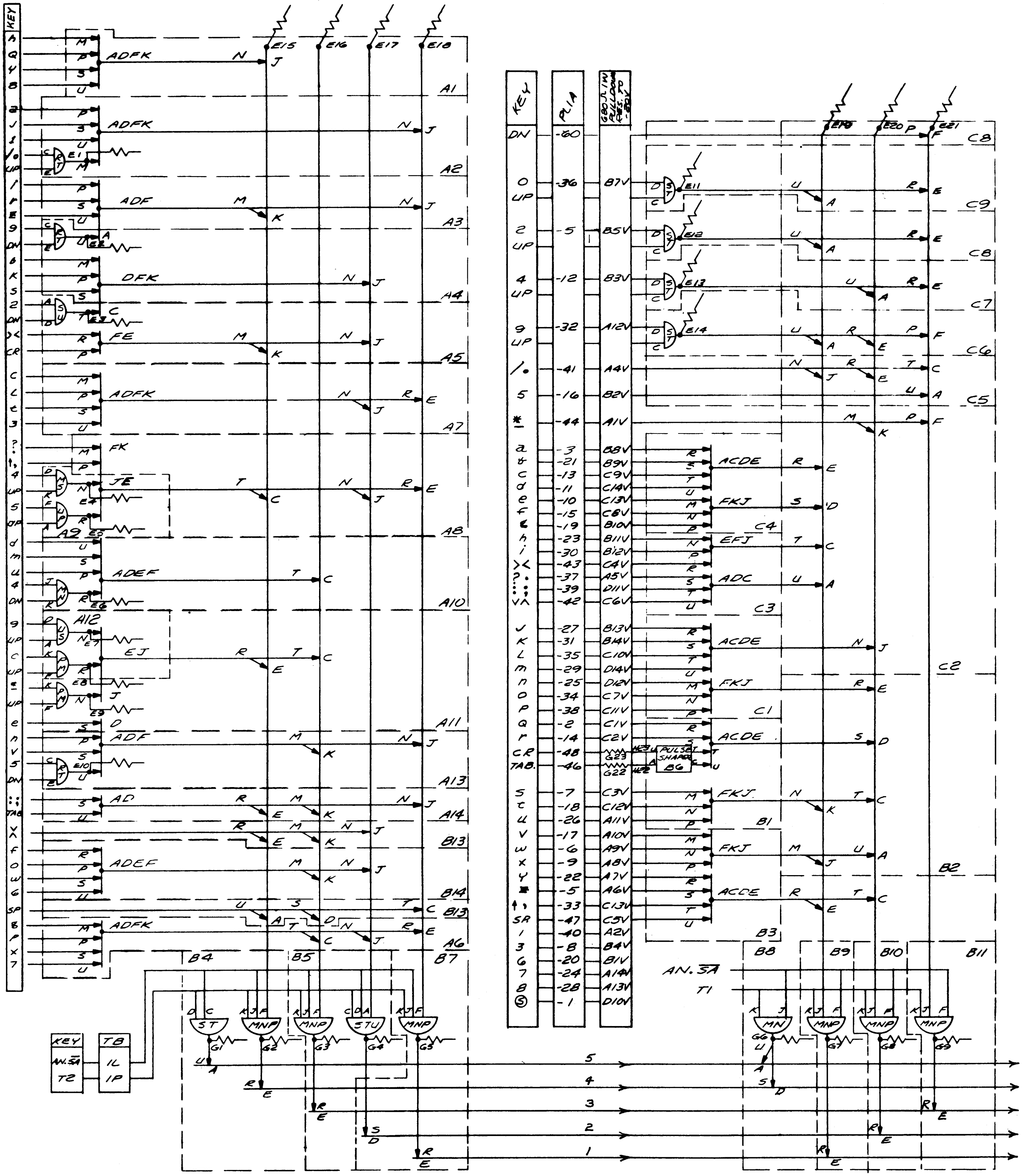
SIGNAL TO TYPEWRITER	TB	PL2A
OV_B	2Z	51
OV_B	1Z	54
OV_B	1F	69



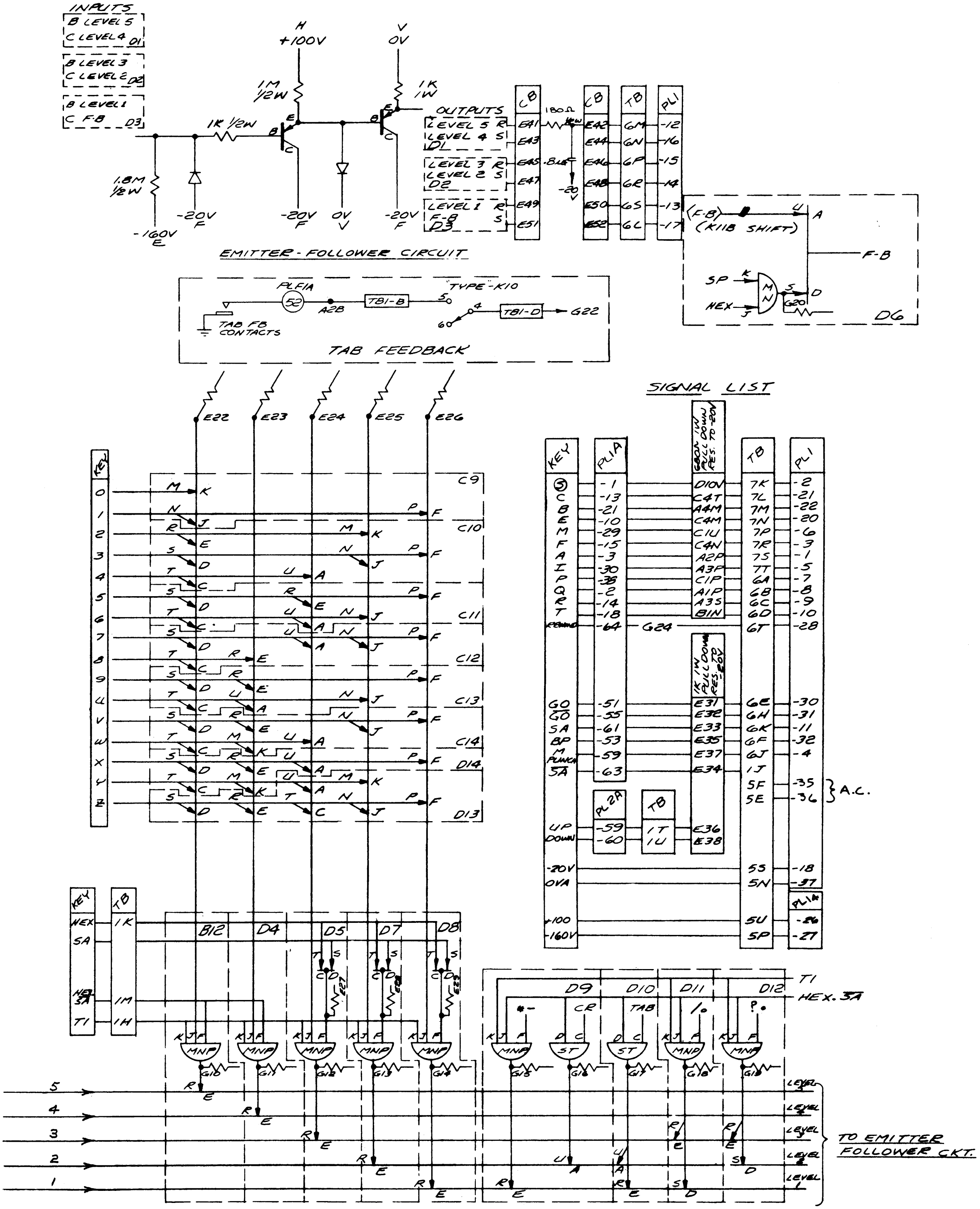
ALL RESISTORS 1/2 W UNLESS NOTED.

NOTES:

SCHEMATIC - DECODER
ANC-1 COUPLER



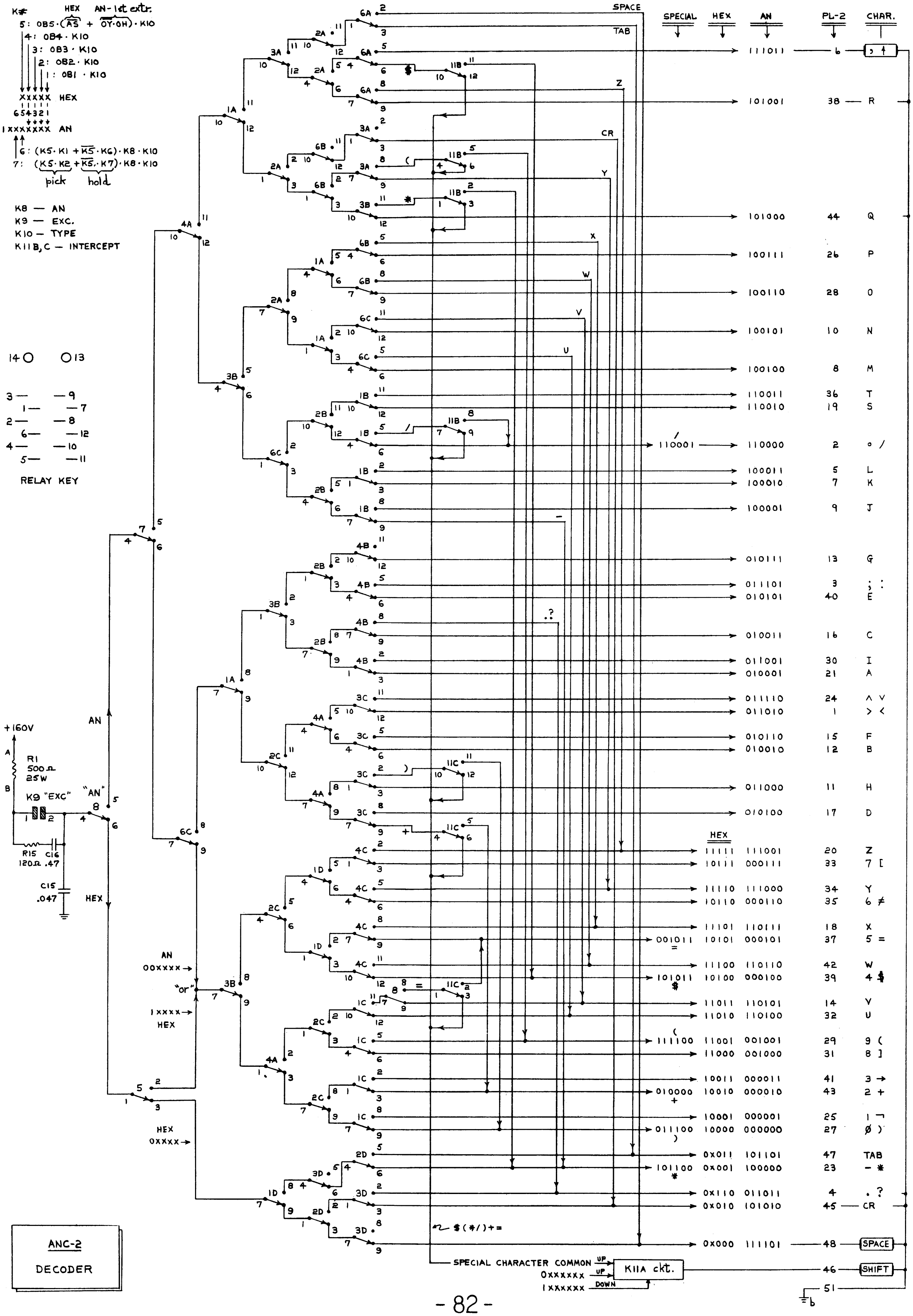
3. RESISTORS "G22" & "G23" ARE 120Ω 1/2W FOR CURRENT LIMITING.
2. RESISTORS "G17", "G18" & "G19" ARE 110K 1W TO +100V.
1. RESISTORS "E1" THRU "E14" ARE 56K 1/2W TO +100V
RESISTORS "E15" THRU "E29" ARE 330K 1/2W TO -160V
RESISTORS "G1" THRU "G16" & "G20" ARE 390K 1/2W TO +100V
NOTES:



SIGNAL LIST

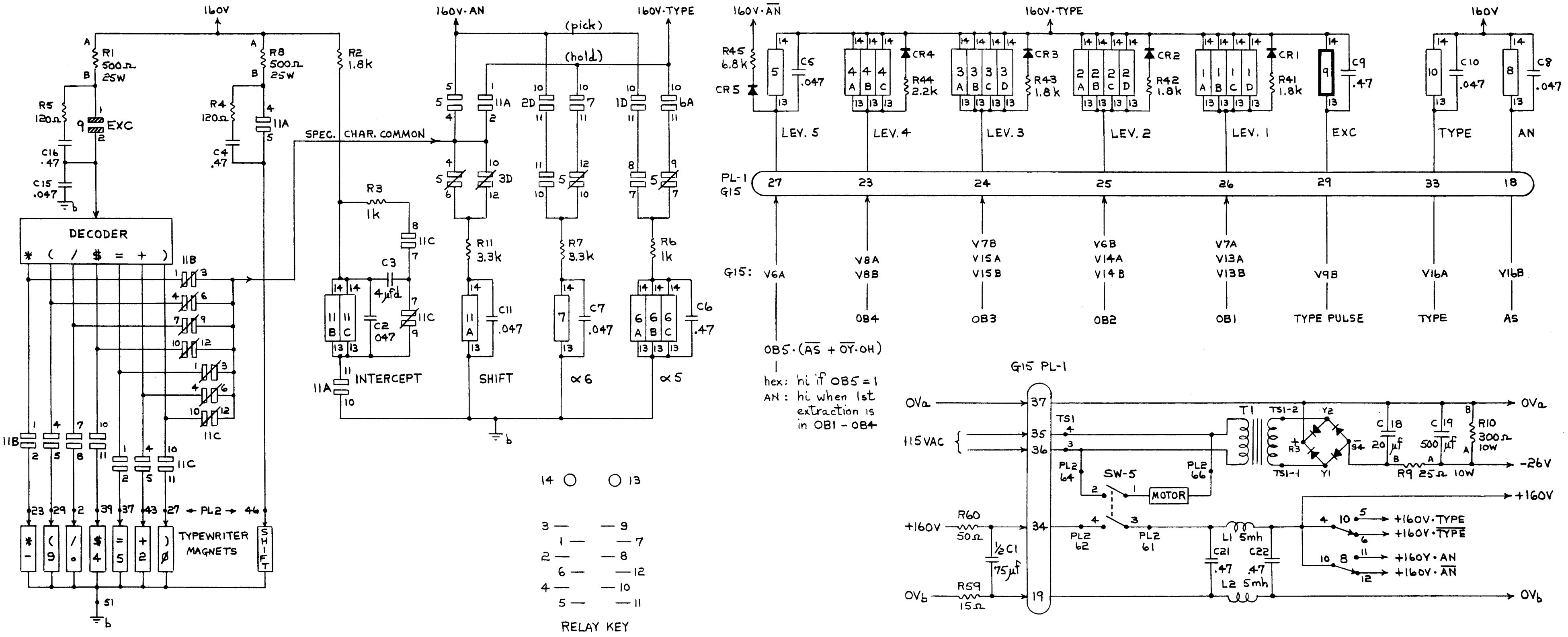
KEY	PL1A	PL2A	TB	680Ω 1W PULL DOWN RES. TO -20V / 1K 1W PULL DOWN RES. TO -20V	TB	PL1
⑤	-1			D10V	7K	-2
C	-13			C4T	7L	-21
B	-21			A4M	7M	-22
E	-10			C4M	7N	-20
M	-29			C1U	7P	-6
F	-15			C4N	7R	-3
A	-3			A2P	7S	-1
I	-30			A3P	7T	-5
P	-35			C1P	6A	-7
Q	-2			A1P	6B	-8
R	-14			A3S	6C	-9
T	-18			B1N	6D	-10
REWIND	-64			G24	6T	-28
GO	-51			E31	6E	-30
$\overline{GO}$	-55			E32	6H	-31
SA	-61			E33	6K	-11
BP	-53			E35	6F	-32
M PUNCH	-59			E37	6J	-4
$\overline{5A}$	-63			E34	1J	
					5F	-35 } A.C.
					5E	-36 } A.C.
UP		-59	1T	E36		
DOWN		-60	1U	E38		
-20V					5S	-18
OVA					5N	-37
						PL1A
+100					5U	-26
-160V					5P	-27

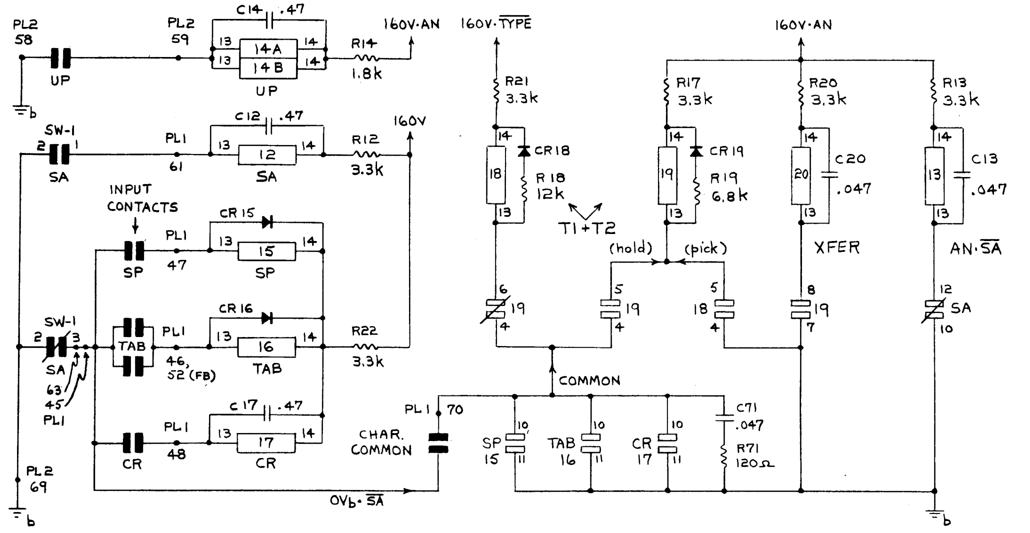
SCHEMATIC - ENCODER
ANC-1 COUPLER



ANC-2

DECODER





K#	FUNCTION	IN		OUT	
		HEX	AN	HEX	AN
1 ABCD	lev. 1	—	—	OB1	OB1
2 ABCD	lev. 2	—	—	OB2	OB2
3 ABCD	lev. 3	—	—	OB3	OB3
4 ABC	lev. 4	—	—	OB4	OB4
5	lev. 5	—	—	OB5	OB5·$\overline{\text{OY}}$·OH
6 ABC	α5	—	—	—	5·1 + [$\overline{5}$·6]
7	α6	—	—	—	5·2 + [$\overline{5}$·7]
8	AN	—	AS	—	AS
9	EXC.	—	—	EXC.	EXC.
10	TYPE	—	—	TYPE	TYPE
11 A	SHIFT	—	—	—	5·$\overline{3}$+SPEC.CHAR.+[$\overline{5\cdot3}$·11A]
11 BC	INT.	—	—	—	11A
12	SA	SA*	SA*	SA*	SA*
13	AN·$\overline{\text{SA}}$	—	AN·$\overline{\text{SA}}$	—	/AN·$\overline{\text{SA}}$/
14 AB	UP	—	UP*	—	UP*
15	SP	SP*·$\overline{\text{SA*}}$	SP*·$\overline{\text{SA*}}$	/SP*·$\overline{\text{SA*}}$/	SP*·$\overline{\text{SA*}}$
16	TAB	TAB*·$\overline{\text{SA*}}$	TAB*·$\overline{\text{SA*}}$	/TAB*·$\overline{\text{SA*}}$/	TAB*·$\overline{\text{SA*}}$
17	CR	CR*·$\overline{\text{SA*}}$	CR*·$\overline{\text{SA*}}$	/CR*·$\overline{\text{SA*}}$/	CR*·$\overline{\text{SA*}}$
18	}T1+T2	COMM·$\overline{\text{SA*}}$	COMM·$\overline{\text{SA*}}$·$\overline{19}$	—	—
19		—	18 + [COMM·$\overline{\text{SA*}}$·19]	—	—
20	XFER	—	19	—	—
	T1	18	18·$\overline{19}$	—	—
	T2	—	$\overline{18}$·19	—	—

* means SWITCH

[xxx] means HOLD

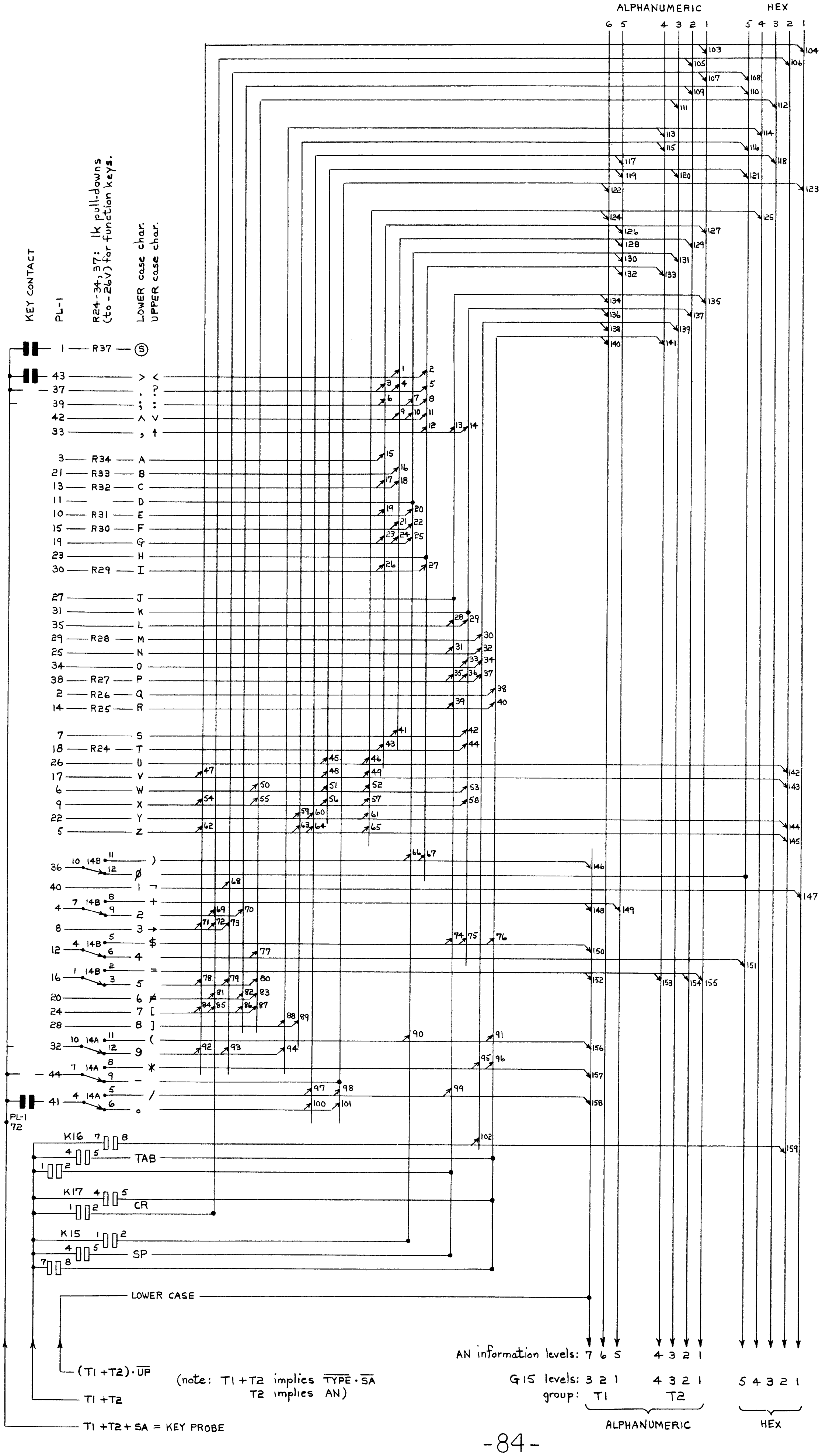
/xxx/ means PRESENCE NOT REQUIRED

COMM = CHAR. COMMON*+ SP+TAB+CR

SPEC. CHAR. = 9·$\overline{\text{11BC}}$·*(/$=+)

ANC-2

RELAY CONTROL

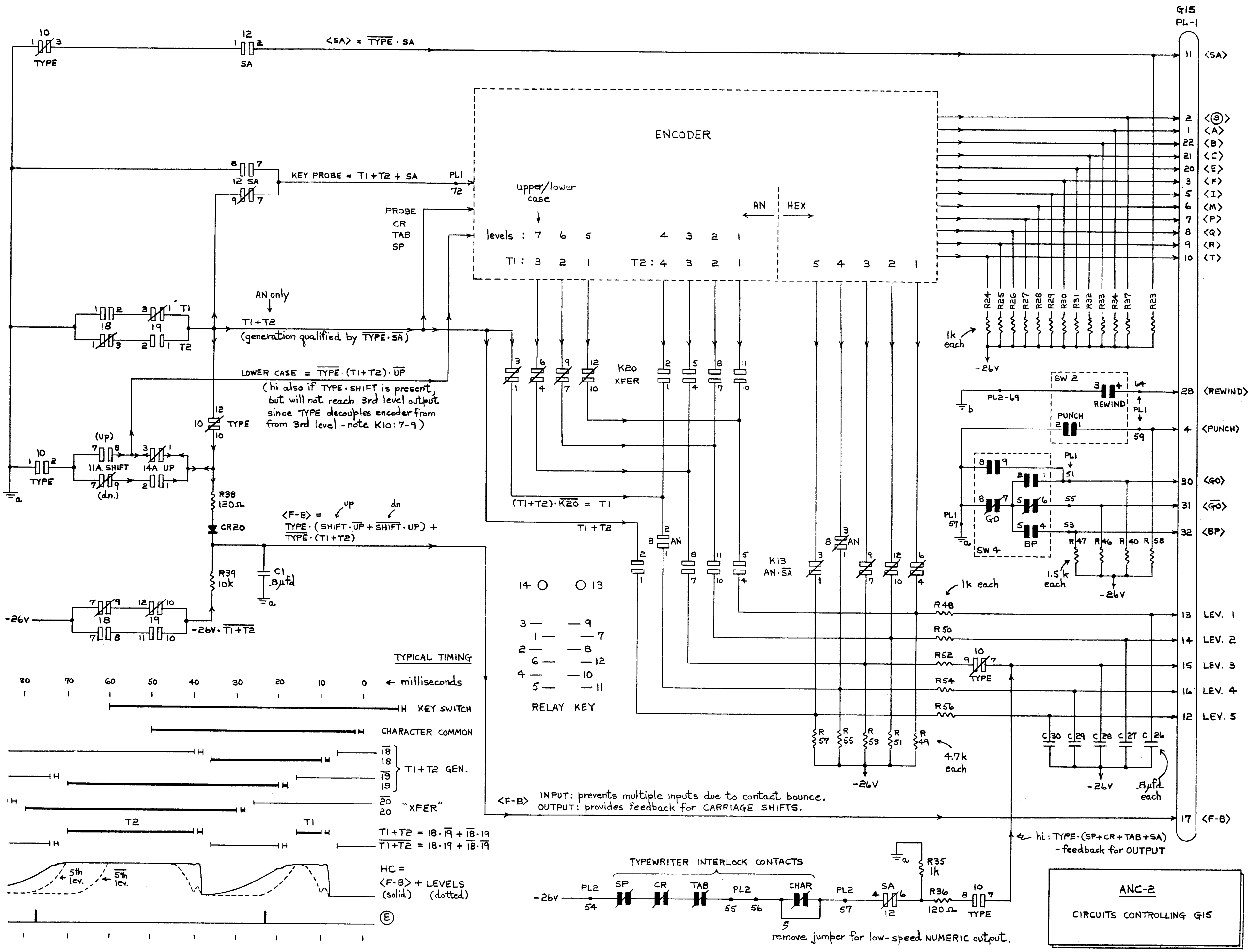


DIODE LOCATIONS

#	pkg.	▶|	#	pkg.	▶|
			80	16	PF
1	12	UA	81	1	PF
2	12	TC	82	1	NJ
3	11	PF	83	1	MK
4	11	NJ	84	1	UA
5	11	MK	85	1	TC
6	11	UA	86	1	SD
7	11	Tc	87	1	RE
8	11	SD	88	15	RE
9	10	PF	89	15	SD
10	10	NJ	90	14	RE
11	10	MK	91	14	SD
12	10	TC	92	15	MK
13	10	SD	93	15	NJ
14	10	RE	94	15	PF
15	10	UA	95	14	MK
16	9	MK	96	14	NJ
17	9	PF	97	13	MK
18	9	NJ	98	13	NJ
19	9	SD	99	13	PF
20	9	RE	100	13	TC
21	9	UA	101	13	UA
22	9	Tc	102	13	SD
23	6	PF	103	23	SD
24	6	NJ	104	23	TC
25	6	MK	105	23	RE
26	8	NJ	106	23	NJ
27	8	MK	107	22	PF
28	8	RE	108	22	RE
29	8	PF	109	22	NJ
30	8	SD	110	22	SD
31	8	UA	111	14	UA
32	8	TC	112	23	PF
33	7	NJ	113	12	SD
34	7	MK	114	22	MK
35	5	PF	115	12	RE
36	5	NJ	116	22	TC
37	5	MK	117	20	MK
38	7	PF	118	21	UA
39	7	SD	119	20	SD
40	7	RE	120	19	PF
41	7	UA	121	22	UA
42	7	TC	122	21	TC
43	4	NJ	123	23	UA
44	4	MK	124	21	SD
45	3	PF	125	20	UA
46	3	NJ	126	20	Tc
47	6	UA	127	19	Tc
48	6	TC	128	20	RE
49	6	SD	129	19	RE
50	5	UA	130	20	NJ
51	5	TC	131	19	NJ
52	5	SD	132	20	PF
53	5	RE	133	12	PF
54	4	UA	134	21	RE
55	4	Tc	135	19	UA
56	4	SD	136	21	PF
57	4	RE	137	19	SD
58	4	PF	138	21	NJ
59	3	UA	139	19	MK
60	3	TC	140	21	MK
61	3	SD	141	12	NJ
62	2	SD	142	3	MK
63	2	RE	143	6	RE
64	2	PF	144	3	RE
65	2	NJ	145	2	MK
66	17	SD	146	17	UA
67	17	TC	147	2	TC
68	2	UA	148	17	PF
69	15	TC	149	17	RE
70	15	UA	150	18	UA
71	18	MK	151	17	NJ
72	18	NJ	152	16	RE
73	18	PF	153	16	SD
74	18	RE	154	16	TC
75	18	SD	155	16	UA
76	18	TC	156	14	TC
77	17	MK	157	14	PF
78	16	MK	158	13	RE
79	16	NJ	159	23	MK

ANC-2

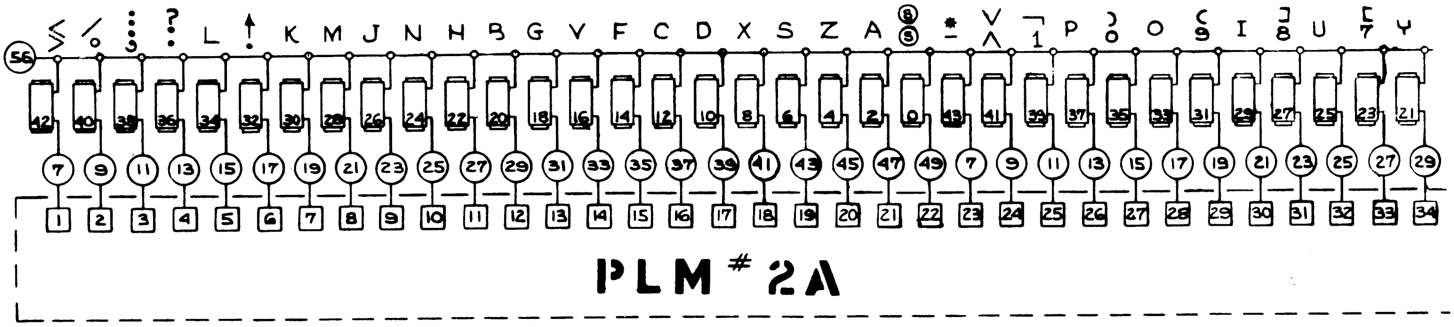
ENCODER

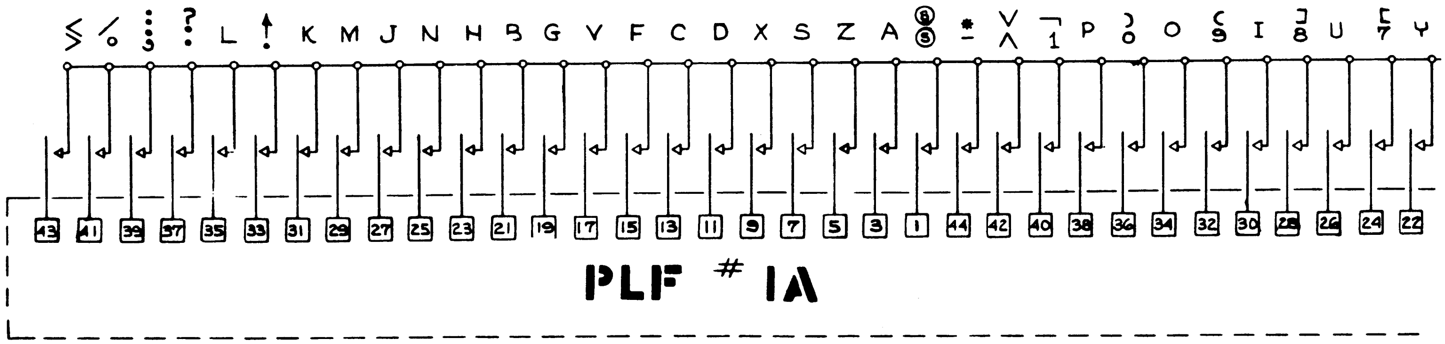


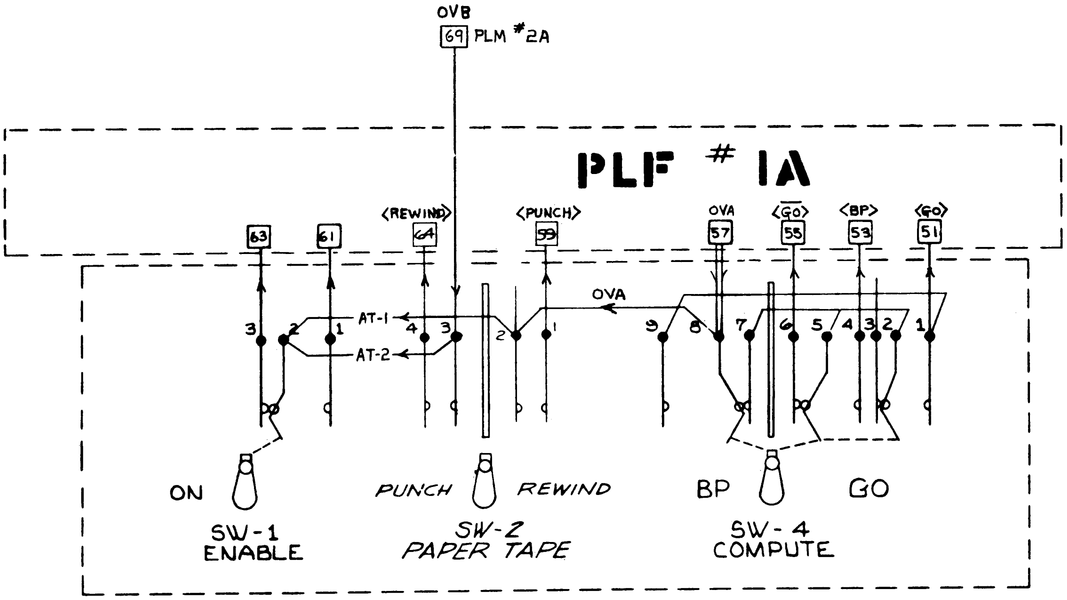
INPUT: prevents multiple inputs due to contact bounce.
OUTPUT: provides feedback for CARRIAGE SHIFTS.

ANC-2

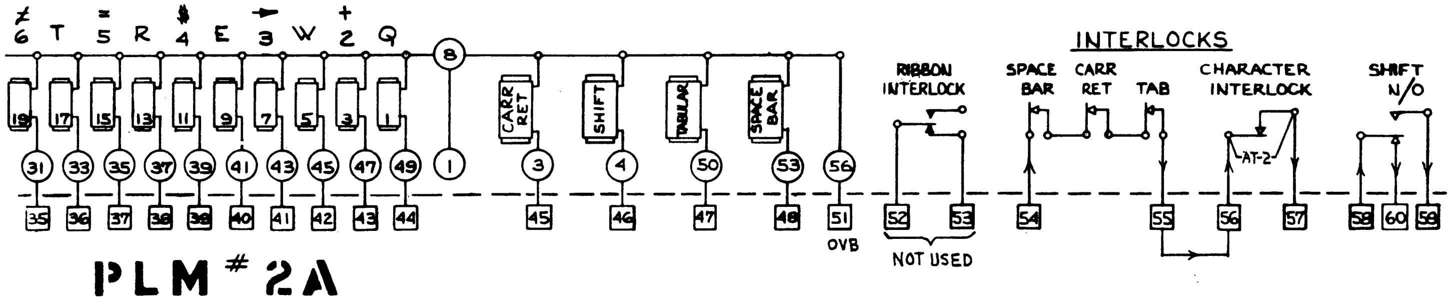
CIRCUITS CONTROLLING G15

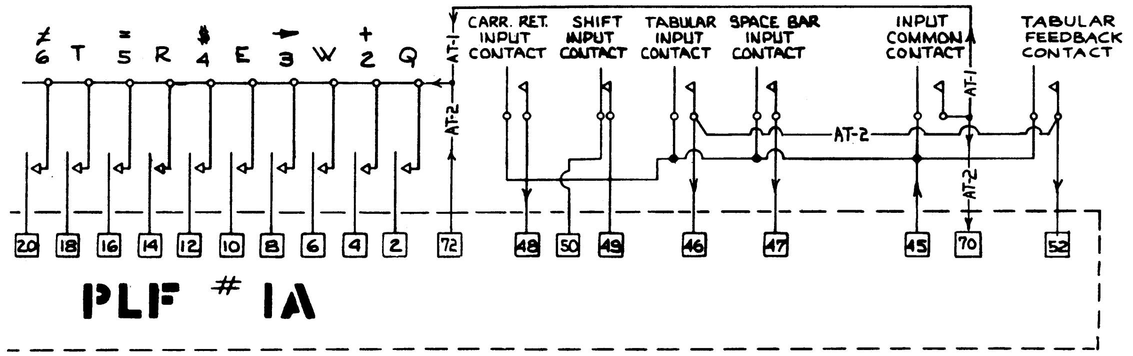


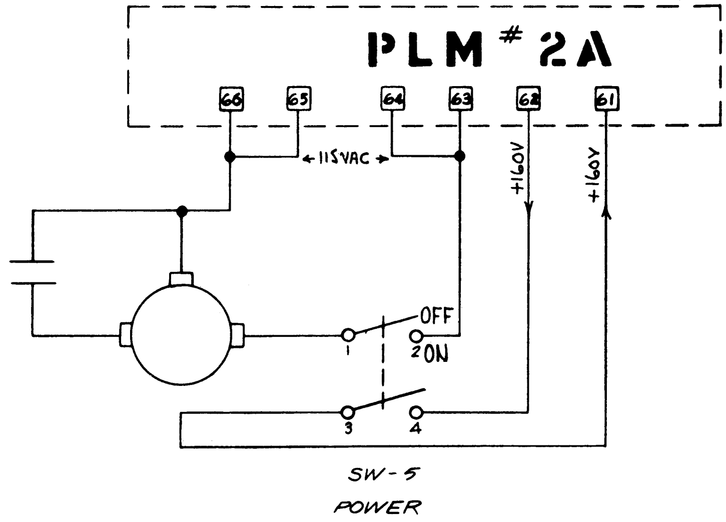




NOTE: An INPUT-OUTPUT WRITER used with an ANC-1 or NC-1 is an AT-1; that used with an ANC-2 is an AT-2. There are minor wiring differences involving the ENABLE SWITCH, CHARACTER INTERLOCK, TAB CONTACTS, and INPUT COMMON as indicated on this drawing. (e.g. •——— AT-2 ———• means the wire exists in an AT-2 but not in an AT-1.) Conversion of an AT-1 to an AT-2 is outlined on dwg. 1D3001.

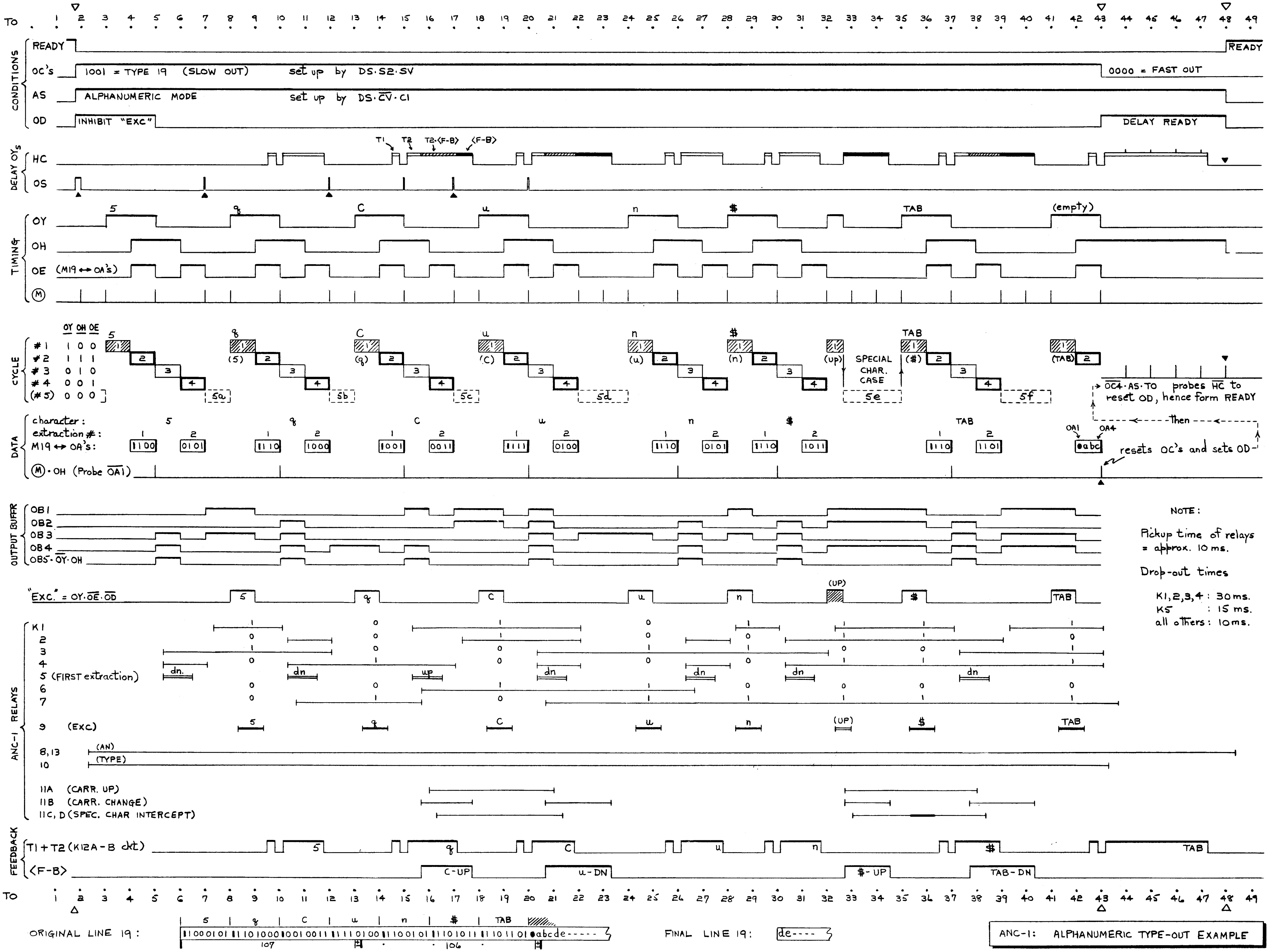






MISCELLANEOUS CONNECTIONS

	ANC-1	ANC-2
PLM2A		
54	OVB	-26V
57	R18	K12-4
58	OVA	OVB
59	ENCODER	K14-13
60	ENCODER	(nc)
PLF1A		
45	OVA	OVB·$\overline{SA}$
46	ENCODER	K16-13
47	ENCODER	K15-13
48	ENCODER	K17-13
49	R10	(nc)
50	K11B-14	(nc)
52	K10-5	(nc)
61	E33	K12-13
63	E34	PLM1A-45
70	K19-4--	(nc)
72	(nc)	T1+T2+SA



ANC-1: ALPHANUMERIC TYPE-OUT EXAMPLE

PART III - APPENDIX

CONTENTS

TEST ROUTINES NO. 1 AND NO. 2

Test Routines No. 1 and No. 2 are combined in one master magazine and a Block Selector Routine is incorporated for selection of the desired test. Test No. 1 is a cursory check of a computer, resulting in a bell being rung at regular intervals. Basically, re-circulation, arithmetic circuits, AR and PN registers are tested by this routine. In order to subject the computer to a more comprehensive test, Test Routine No. 2 should be used.

It is assumed that the input circuits (including the pertinent memory lines - L. 19, L. 23 and L. MZ), L. 00, the command and control circuits are in working condition. If one or more of these is making errors, the result will be erratic operation, usually evidenced by the computer getting lost in a loop of commands.

METHOD OF OPERATION

1. Read the Number Track in - p key.

2. Read in the Block Selector Routine by striking p key again.
   (Steps 1 and 2 must be executed when machine is ON and are automatically executed during the start cycle.)

3. Set Compute switch to GO - a type-out will occur.

4. Type in 0000002 tabⓢfor Test Number One. Test Number One will then begin: The first block of tape (the loader routine for Test No. 1) is read into the computer. If it is properly recorded, the type-out will be: XXXXXXX. The second block of tape is then automatically read in and terminated when the bell rings twice. (There is no type-out unless an error is encountered.) The second block of tape then forms a closed loop in the computer and is repeated indefinitely until the GO-BP switch is moved to the center position.

5. If Test Number Two is desired, type in 0000005 tabⓢand perform steps 6 and 7.

6. The computer will stop on a Test Ready, at which time manual type-in will have been set. The operator must then type a sexadecimal number which is the number of times each type of test will be executed. For example, if 0000010 is typed,

each line of the memory will be tested 16 times before the next type of test is accomplished. The normal method of seven digits, tab, and Ⓢ key is the method used to type this number.

7. After the Ⓢ key is hit, the computer will proceed to execute each type of test without further manual intervention (unless the computer gets lost). The first test made is a test to determine if the accumulator and its associated circuits work reliably. The test involves adding all the commands in Line 0 into the accumulator and subtracting them out again. If the AR is clear after this, the AR is considered reliable. If it is not clear, a number is typed out which is characteristic of this test. (See the list of indications.)

   In accomplishing the above test, Line 23 is considered to be reliable and is used to store numbers for determination of the end of the test. A bell is rung at the end of the test.

8. The second test is the determination of the reliability of the two-word registers. Again, if an error is made, a characteristic number is typed. Certain four-word registers and the AR are used in this test and a bell rung upon its completion.

9. The four-word registers are then tested and characteristic numbers typed if an error is made. The two-word registers and AR are used in this test. A bell is rung at the completion of this test also -- the third and last bell in the first block of commands.

10. The next block of commands are then read in. This block tests all of the long lines from one to nineteen by putting the same information as is in line zero into all of them. It fills each line only once at the beginning of this routine unless a certain line makes an error, at which time the information in that line is renewed. As a result, the successful completion of this test is assurance that all lines have stored information for a considerable period of time, i. e., equal to the total length of time of the entire test.

    This method of indication of errors results in a minimum of type-out. A type-out is not made each time a line makes an error. Instead, the number of errors made by each line is counted and this number typed at the completion of the entire test. At the completion of the test, then, a type-out is made; if no errors have been made, only a series of spaces are typed and nothing will appear on the printed page.

However, if Line 9 has made fifteen errors (this could or could not be the total number of tests made on each line), a number will be typed out as follows:

690000z

Note that the second digit is the line number and the last digit is the number of errors (in sexadecimal). If the number is 7300012, the line would be Line 19 and it would have made 18 errors.

11. The next block of commands is then read in. This tests the inverting gates and the sign circuits of the two-word registers. Two bells are rung, one.after each of the above-named tests.

NOTE

> If the inverting gates are not working, chances are that the tests made in the preceding blocks did not work and have given false indications. The inverting gate test is used to determine if the error indications were caused by the inverting gates.

12. The next block, after being read in, will test the overflow circuits and all combinations of End-Around-Carry conditions in both the AR and PN registers. One bell is rung upon completion of both tests and characteristic numbers are typed in case of error.

13. The next block of commands has three types of tests in it. The first is a multiplication and division test. It involves the determination of the equality:

$$A = \frac{A \cdot B}{B}$$

If this equality is not met, the computer then determines if the equality $A \cdot B = B \cdot A$ is met. ($A \cdot B$ is the same multiplication made for testing the first equality.) If this equality is not met, the computer types out a number characteristic of a multiplication error, while if it is met, a division error indication is made. A bell is rung upon completion of the designated number of tests.

14. The second test is of the shifting and normalizing circuits. The test normalizes a number and then shifts it back where it was, hence, compares it with the starting number. A bell is rung upon completion of this test.

15. The third test is of the logical commands. The third bell is rung here.

16. The next block of commands results in two types of tests. The first is a type-out of six test numbers in the following sequence:

    ```
    -1122334        445566.7        778899
    -uuvvwwx        xxyyzz.0          2345
    ```

17. The second test in the last block is a series of computations using two standard subroutines. The blocks of commands for the subroutines are read in by the last test block before the type-out of the above test numbers.

    The computations involve the calculation of the sine of an angle θ and the calculation of the arcsine of sine 0 to produce an angle ϕ. If the computations are correct, $\phi = \theta$, thereby checking this equality. If the angles are not equal, a characteristic number is typed out. The angle θ is then varied by an incremental amount and the computation repeated the number of times specified by the operator. The test type-out will occur during some of these computations.

18. At the end of computations, the tape automatically reverses to the beginning of the test routine and the operation started at the first test in number 4 above. If the operator wishes to change the number of times each test is performed, he should stop computations when the first block is being read in and then do the operations starting with number 1.

ERROR INDICATIONS FOR TEST ROUTINE NO. 1

Typed Number	Likely Error	Possible, But Not Likely
159539v	Inverting gates (IG), LI	AR
21u139v	LII	LI AR IG
33v339v	PN	AR IG
3xvx75v	ID MQ	PN* IG
-6466w9y	L4	PN* IG LII
-6466wvy	L5	PN* IG LII
-6466wxy	L6	PN* IG LII
-6466wzy	L7	PN* IG LII
-6466xly	L8	PN* IG LII
-6466x3y	L9	PN* IG LII
-6466x5y	L10	PN* IG LII
-6466x7y	L11	PN* IG LII
-6466x9y	L12	PN* IG LII
-6466xvy	L13	PN* IG LII
-6466xxy	L14	PN* IG LII
-6466xzy	L15	PN* IG LII
-6466yly	L16	PN* IG LII
-6466y3y	L17	PN* IG LII
-6466y5y	L18	PN* IG LII

*See Discussion of PN, Test Routine No. 1: Memory Test Indications

Typed Number	Likely Error	Possible, But Not Likely
5300	Overflow flip-flop didn't sense OF. Test overflow didn't work.	Program or operator's error. **
56000	Overflow FF not turned off by test.	Test circuit not working properly.
6000	Source 30 or wl	LI LII
12000	Source 27, 29	LI LII
1x000	Multiplication Division LIII	PN** or IG
58000	Shift or Normalize	ID, MQ
32000	Input circuits Photo reader LV, L19, LIV	

**See Discussion of Test Routine No. 2: Typed Indications of Errors

ERROR INDICATIONS FOR TEST ROUTINE NO. 2

Various characteristic numbers are typed out whenever the computer makes an error. The following is a list of these numbers, the type of error made, and some possibilities as to computer circuits which may have made the error.

1. Accumulator Register Test

393939v This test does additions and subtractions of the commands in LO in the accumulator. If an error is made, it is usually an indication that the accumulator is not holding information reliably, but other circuits such as the inverting gates may also be operating improperly.

Look for:

1. Accumulator Read head out of tangential adjustment.
2. Bad AR Read amplifier.
3. Weak tube in AR preamplifier.
4. Bad AR record amplifier.
5. Bad component in accumulator adder.
6. Bad component in inverting gates. (Try replacing IS flip-flop)
7. Read or Write head open or shorted.
8. Bad taper pin connection between logic panel and memory.
9. Source or destination selector gates bad.

2. PN Register Test

555539v This test stores data in the PN register and compares it against data stored in Line 23. Therefore, Line 23 may be making the error, but since Line 23 was used for read-in, one suspects PN. The AR is used for this test so it may also be making the error, although it should be okay if it passes its test.

Look for:

1. PN register Read head out of tangential adjustment.
2. Weak Read amplifier.
3. Weak preamplifier tube.
4. Bad record amplifier.
5. Bad component in PN adder and associated circuitry.
6. Read or Write head open or shorted.

7. Bad taper pin connection to or from memory.
8. Source or destination selector gates bad.

3. <u>ID and MQ Register Tests</u>

656839v ID }
y0y39v MQ } See discussion of PN test.

Look for:

1. Read head out of tangential adjustment.
2. Bad Read amplifier.
3. Bad preamplifier tube.
4. Bad record amplifier.
5. Recirculation, source or destination gates bad.
6. Read or Write head open or shorted.
7. Bad taper pin connection.

4. <u>Lines 20, 21, 22, 23</u>

616139v - L20
x1x39v - L21
454639v - L22
595w39v - L23 The AR and two-word registers are used in this test. They could be making the error even though they passed their test, but this is unlikely. If L23 indication is made, careful scrutiny is in order because it worked properly to read the data from tape.

Look for:
Same things as listed for ID and MQ Registers.

5. <u>Long Line Tests</u>

730000n - Line 19
720000n - Line 18
710000n - Line 17
700000n - Line 16
6z0000n - Line 15
6y0000n - Line 14
6x0000n - Line 13
6w0000n - Line 12

6v0000n - Line 11
6u0000n - Line 10
690000n - Line 9
680000n - Line 8
6-------------

610000n - Line 1
The number N is the number of times the particular line makes an error. Short lines are used in the test, so it should be remembered that they can make errors even though they have passed their own tests. This is unlikely, however.

Look for:
Same things as listed for ID and MQ Registers.

6. Miscellaneous Test

Continuous Bell Ringing
The Long Lines test requires the storage of test data in Line 23. If this data becomes in error, the computer rings a bell continuously. To stop the bell-ringing, hit the Ⓢ key, at which time the computer will try again. However, since Line 23 or some other circuit not connected with a long line failed, the best thing to do is to start the whole routine over to test the other circuits again.

Causes may be:
1. Line 23 unreliable
2. PN register unreliable

7. Inverting Gates

222439v Additions and subtractions of very simple numbers into the AR are not working properly. The numbers are stored in lines 21 and 22, so these lines could have failed.

Look for:
1. Bad IS flip-flop.
2. Bad IC flip-flop.
3. Other bad component in the inverting gates.
4. Bad AR (See AR test).
5. Bad Line 21 or Line 22 (See test number 4).

8. Sign Circuits of Product Registers

1y1y2xz Sign did not properly set up to be negative when a negative number when a negative number entered the ID and a positive entered the MQ.

2x2x29w Sign was negative when two positive numbers were entered into ID and MQ.

3w3w2xz Sign was positive when a positive number was entered into the ID and a negative number into the MQ.

484w2xz Sign was negative when two negative numbers were entered into ID and MQ.

9. Reader Test

nnnnnnn

5w5w39v When reading in the next block of commands, the computer checks the read-in for errors. It types out the block sum that was obtained (which should have been zero) and the indication number.

Look for:

1. Photo reader out of adjustment.
2. Bad Line 19 or 23.
3. Bad AR register (See AR test).
4. Other bad component in input circuitry.

10. Overflow Test

Two quantities, A and B, are used in this test. A is equal to 1/2, B is slightly greater than 1/2.

32323vz Overflow did not set when A + B was added in the AR.

3u3u3vz Overflow did not set when -A -B was calculated in AR.

44443vz Overflow did not set when -A -A was calculated in AR.

4y4x3vz Overflow set up when the difference of two positive numbers was taken in AR.

5u593vz Overflow set up when the difference of two positive numbers was taken in PN.

66663vz Overflow did not set up when A + B was added in PN.

u0u3vz Overflow did not set up when -A -B was calculated in PN.

191u3vz Overflow did not set up when -A -A was calculated in PN.

11. End-Around-Carry Test

This subtracts zero from AR and PN to see if the End-Around-Carry is propagated to correct the sign.

25262xz The sign of (-A -0) was positive, indicating no End-Around-Carry when calculated in AR.

363v2xz The sign of (B - 0) was negative when calculated in PN.

12. Multiplication - Division Test

Two numbers, A and B, are used in this test. The identity $A \cdot B = C$ is assumed in the discussion.

z5v294 The equality $A = \frac{C}{B}$ does not check, but $A \cdot B = B \cdot A$ does check. This indicates division failed.

Look for:

1. Bad IS flip-flop.
2. Bad IC flip-flop.
3. Sloppy wave forms on input to IS flip-flop or elsewhere in inverting gates.
4. Bad buffer-inverter driving "division" signal.
5. Bad component in PN adder.
6. A bad two-word register.

105v2v5 The equality $A = \frac{C}{B}$ does not check nor does the equality $A \cdot B = B \cdot A$. Lines 20 and 23 are used to store intermediate data in this test.

Look for:

1. Bad PN flip-flop or gate driving this flip-flop.
2. Bad PN adder.
3. Slow rising or falling signals in adder inputs or output.
4. Bad two-word register.
5. Bad Line 20 or Line 23.

13. Shift and Normalize

A number is put in the even half of MQ and normalized. After normalization, it is put in ID and shifted right, with the number of shifts determined by the number accumulated in the AR when normalization occurred.

The number should end up in the same position of ID that it started in MQ.

65662xw The number left in ID at the end of the shifting operation does not agree with the original number which started in the MQ.

Look for:

1. Bad ID or MQ register.
2. Bad AR register.
3. Circuits used to turn on AR carry for incrementing not working properly.
4. Bad PN flip-flop or associated circuits.
5. Gates controlling control circuits on normalize or on shifting not working properly.

14632zz This is a miscellaneous test. Upon shifting the ID right, the MQ should also be shifted left. Also, the number of shifts should be sufficient to shift all the contents of MQ off the left-hand end, clearing MQ. This number will be typed out if MQ is not clear.

Look for:

1. Bad MQ register.
2. Same things as listed on test above.

14. Logical Commands

x0513vw One or both of the logical commands $20\cdot 21$ and $\overline{20}\cdot 21$ (Sources 31 and 30) did not work properly.

Look for:

1. Bad Line 20 or 21.
2. Bad gates for logical commands.

u4292xw The command $20\cdot 21 + \overline{20}\cdot \text{AR}$ (Source 27) did not work properly.

Look for:

Same as above.

15. Test Type-Out

A test type-out occurs which operates all possible characters of the typewriter:

```
-1122334       445566.7       778899
-uuvvwwx       xxyyxx.0         2345
```

If this does not type correctly,

Look for:

1. Sticking keys on typewriter.
2. Broken or sticking relays in typewriter base.
3. Bad OB flip-flop.
4. Bad Line 23, 19 or 2.
5. Any other Input/Output circuit.

16. Computation Test

w7483zv This test involves the calculation of the sine of an angle θ and the calculation of the arcsine of this result. If the arcsine agrees with θ, the test is passed. This is the last test of the routine and since practically all circuits are tested before this, should work. If not,

Look for:

1. Bad CD flip-flop (this was not tested before) or circuit driving CD.
2. Something wrong with computation registers (all short tracks).
3. Since type-out occurs during computation, look for arcing contacts in typewriter relays.
4. Any other portion of machine bad.

SUMMARY OF ERROR INDICATIONS

1.	393939v	Accumulator Register		
2.	555539v	PN Register		
3.	656839v	IO Register		
4.	y0y39v	MQ Register		
5.	616139v	Line 20		
6.	x1x39v	Line 21		
7.	454639v	Line 22		
8.	595w39v	Line 23		
9.	6?0000N	Line ?		
10.	7?0000N	Line 16 + ?		
11.	Continuous Bells	Line 23, or PN		
12.	222439v	Inverting Gates		
13.	lyly2xz	Sign Circuits		
14.	2x2x29w	Sign Circuits		
15.	3w3w2xz	Sign Circuits		
16.	484w2xz	Sign Circuits		
17.	5w5w39v	Input Error		
18.	32323vz	Overflow Circuits		
19.	3u3u3vz	Overflow Circuits		
20.	44443vz	Overflow Circuits		
21.	4y4x3vz	Overflow Circuits		
22.	54593vz	Overflow Circuits		
23.	u0u3vz	Overflow Circuits		
24.	191u3vz	Overflow Circuits		
25.	25262xz	End-Around-Carry - AR		
26.	363v2xz	End-Around-Carry - PN		
27.	z5v294	Division		
28.	105v2v5	Multiplication		
29.	65662xw	Shift or Normalize		
30.	14632zz	Shift or Normalize		
31.	x0513vw	Source 30 or 31		
32.	u4292xw	Source 27		
33.	-1122334	445566.7	778899	Test Type-Out
	-uuvvwwx	xxyyzz.0	2345	
34.	w7483zv	Computational Error		

TEST ROUTINE NO. 1 Line 00

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
u0	u1	u2	u3
u4	u5	u6	

L	P	T or L_k	N	C	S	D	BP	NOTES
00		02	00	0	28	31		Test Ready
01		02	03	0	00	28		Clear AR
03	u	04	04	1	00	29		Add all L0
04	u	09	09	1	28	20		AR → 20-D, 1, 2, 3
09		10	11	0	00	28		Clear AR
11	u	12	12	1	00	29		Add all L0
12	u	17	17	1	28	21		AR → 21-0, 1, 2, 3
17		18	19	3	20	29		AR -(20 -2) → AR
19		20	21	0	28	27	-	Test AR Bk. Pt.
21	u	26	26	1	20	29		Add all L 20
22	-	19	u0	0	00	28		(00 - 19) → AR
26	u	31	31	3	21	29		Subt. all L 21
31		32	33	0	28	27	-	Test AR Bk. Pt.
33		34	35	0	20	28		(20 - 02) → AR
34		31	u0	0	00	28		(00 - 31) → AR
35	u	38	38	1	28	26		AR → PN (e + o)
38	u	41	41	1	28	25		AR → ID (e + o)
41	u	44	44	1	28	24		AR → MQ (e + o)
44		45	47	1	26	28		PN odd → AR
47		48	49	3	26	29		AR - (PN even) → AR
49		50	51	0	28	27	-	Test AR Bk. Pt.
51		52	54	5	25	26		(ID) → PN
52		49	u0	0	00	28		(00 - 49) → AR
54		56	58	7	24	30		PN - (MQ) → PN
58	u	61	61	1	26	27		Test PN Bk. Pt.
61	u	62	63	5	00	30		Add all L0 → PN

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
u0	u1	u2	u3
u4	u5	u6	

L	P	T or L_k	N	C	S	D	BP	NOTES
62		58	u0	0	00	28		(00 - 58) → AR
63		64	66	5	26	21		PN → 21 - 0, 1
66		47	71	0	00	21		Command 2 → 21 -3
71		74	76	0	00	21		Command 1 → 21 -2
76		78	79	0	21	28		(21 -2) → AR
79		80	81	0	00	29		AR + "1_d" → AR
81		82	83	0	28	21		AR → 21 -2
83		85	85	0	31	31		Obey AR
(8 74	5)u	86	86	0	00	03		L0 → LX
86		88	90	5	21	26		(21 -0, 1) → PN
90		91	92	0	21	28		(21-3) → AR
92		93	94	0	00	29		AR + "1_s" → AR
94		95	96	0	28	21		(AR) → 21 -3
96		98	99	0	31	31		Obey AR
(99)	u	u0	u2	7	03	30		Subtract all LX → PN
u2		u4	05	5	26	27	-	Test PN Bk. Pt.
05		06	08	0	21	28		21 -2 → AR
06		08	06	0	28	31		Test Ready
08		13	14	3	00	29		AR - "18_d" → AR
14		15	75	0	22	31		Test AR sign
75		76	77	0	17	31		Ring Bell
u0		u2	u0	0	28	31		Test Ready
u1		u3	00	0	08	31		Type AR
77		79	79	1	21	31		Transfer to L1 com.
07		09	15	0	08	31		Type AR
15		17	15	0	28	31		Test Ready

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	16		17	05	0	28	28		"Do Nothing"
8	9	10	11	80	u	00	00	0	00	01		"1_d"
12	13	14	15	93	u	00	00	0	01	00		"1_s"
16	17	18	19	13	u	86	86	0	00	18		"18_d"
20	21	22	23									
24	25	26	27									
28	29	30	31									
32	33	34	35									
36	37	38	39									
40	41	42	43									
44	45	46	47									
48	49	50	51									
52	53	54	55									
56	57	58	59									
60	61	62	63									
64	65	66	67									
68	69	70	71									
72	73	74	75									
76	77	78	79									
80	81	82	83									
84	85	86	87									
88	89	90	91									
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

TEST ROUTINE NO. 1 Line 01

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
u0	u1	u2	u3
u4	u5	u6	

L	P	T or L_k	N	C	S	D	BP	NOTES
00		02	00	0	28	31		Extra commands for
01		02	79	0	28	28		convenience of operator
79		81	81	0	15	31		Read photo tape
81		83	83	0	29	31		Test overflow
83		81	56	0	01	28		01_{81} → AR
56		57	58	0	28	21		AR → 21_{01}
58		61	62	0	01	20		E → 20_{01}
62		65	66	0	31	28		(20. 21 - 1) → AR
66		69	70	0	28	21		AR → 21_{01}
70	w	01	70	0	28	31		Test ready
71		73	72	0	08	31		Type AR
72	w	01	72	0	28	31		Test ready
73		74	75	0	01	28		01_{74} → AR
75		77	80	0	21	29		AR + 21_{01} → AR
80		82	82	0	31	31		N. C. from AR
84		86	86	0	29	31	-	Test overflow
86		91	92	4	01	20		01_{91} → 20_{03}
92		95	96	0	01	21		01_{95} → 21_{03}
96		99	u0	0	31	28		20. 21_{03} → AR
u0		u3	u4	0	30	29		AR + 20. 21_{03} → AR
u4		u7	04	3	21	29		AR - 21_{03} → AR
04		05	06	0	28	27	-	Test AR B/P
06		07	08	0	21	28		21_{03} → AR
08		11	12	0	27	28		AR: 0
12		15	16	3	21	29		AR - 21_{03} → AR

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	16		17	18	0	28	27	-	Test AR B/P
8	9	10	11	18		21	21	0	23	31		Clear PN, ID, MQ
12	13	14	15	21		22	23	0	01	20		A → 20_{02}
16	17	18	19	23		24	25	0	01	20		B → 20_{00}
20	21	22	23	25		26	28	6	20	25		A → ID
24	25	26	27	28		32	35	6	20	24		B → MQ
28	29	30	31	35		56	93	0	24	31		Multiply
32	33	34	35	93		94	97	4	26	22		PN → $22_{02,\ 03}$
36	37	38	39	97		u0	u2	6	20	25		B → ID
40	41	42	43	u2		u6	05	4	22	26		$22_{02,\ 03}$ → PN
44	45	46	47	05		v6	14	5	25	31		Divide
48	49	50	51	14		15	17	0	24	22		MQ → 22_{03}
52	53	54	55	17		19	20	1	22	28		22_{03} → AR
56	57	58	59	20		22	26	3	20	29		AR - A → AR
60	61	62	63	26		27	29	0	28	27	-	Test AR B/P
64	65	66	67	29		32	32	0	23	31		Clear MQ, PN, ID
68	69	70	71	32		34	37	0	20	24		A → MQ
72	73	74	75	37		(66)	u5	0	27	31		Normalize MQ
76	77	78	79	u5		u6	09	3	28	28		-AR → AR
80	81	82	83	09		11	15	1	24	25		MQ → ID_1
84	85	86	87	15		(66)	78	0	26	31		Shift
88	89	90	91	78		80	82	0	25	20		(ID even) → 20_{00}
92	93	94	95	82		84	85	1	20	28		20_{00} → AR
96	97	98	99	85		86	88	3	20	29		AR - 20_{02} → AR
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	88		89	98	0	28	27	-	Test AR B/P
8	9	10	11	98		99	55	0	17	31		Ring bell
12	13	14	15	55		57	00	0	20	31		N. C. from L. 00
16	17	18	19	99		85	56	0	01	28		01_{85} → AR
20	21	22	23	87		84	56	0	01	28		01_{84} → AR
24	25	26	27	19		16	56	0	01	28		01_{16} → AR
28	29	30	31	30		26	56	0	01	28		01_{26}→AR
32	33	34	35	07		04	56	0	01	28		01_{04} → AR
36	37	38	39	34		46	46	0	07	31		Reverse photo tape
40	41	42	43	46	u	47	47	1	19	29		Add L. 19 → AR
44	45	46	47	47	u	48	48	3	01	29		Subtract L. 01 → AR
48	49	50	51	48		49	50	0	28	27	-	Test AR B/P
52	53	54	55	50		52	52	0	28	31		Test ready
56	57	58	59	51		48	56	0	01	28		01_{48} → AR
60	61	62	63	52	w	01	52	0	28	31		Test ready
64	65	66	67	53		55	55	0	15	31		Read photo tape
68	69	70	71	40	w	01	40	0	28	31		Test ready
72	73	74	75	41		43	43	0	07	31		Reverse photo tape
76	77	78	79	43		45	43	0	28	31		Test ready
80	81	82	83	44		46	46	0	07	31		Reverse photo tape
84	85	86	87	57		u0	u2	6	20	25		20_{00} → ID_{01}
88	89	90	91	59		u0	u2	6	20	25		20_{00} → ID_{01}
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	60		u0	u2	6	20	25		20_{00} → ID_{01}
8	9	10	11	60		u0	u2	6	20	25		20_{00} → ID_{01}
12	13	14	15	64		u0	u2	6	20	25		20_{00} → ID_{01}
16	17	18	19	65		u0	u2	6	20	25		20_{00} → ID_{01}
20	21	22	23	67		u0	u2	6	20	25		20_{00} → ID_{01}
24	25	26	27	68		u0	u2	6	20	25		20_{00} → ID_{01}
28	29	30	31	69		19	96	0	00	00	-	Go to Loc 96
32	33	34	35	74	u	84	00	3	28	29		Clear AR
36	37	38	39	u7		54	65	7	20	31	-	
40	41	42	43	02								-zy855z4
44	45	46	47	22								-zyxwv10 "A"
48	49	50	51	24								9876543 "B"
52	53	54	55	91								9999999 Extractor for test
56	57	58	59	95								3w3w3w3 Test number
60	61	62	63	61								0072000 E TN Extractor
64	65	66	67									
68	69	70	71									
72	73	74	75									
76	77	78	79									
80	81	82	83									
84	85	86	87									
88	89	90	91									
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

TEST ROUTINE NO. 2 - Short Track Test

Line 0

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	0		1	002	0	19	00		L19 → L0
8	9	10	11	2	-	6	009	0	21	31		Commands L0
12	13	14	15	9		12	023	0	23	31		Halt
16	17	18	19	23		25	027	0	12	31		Set Type-in
20	21	22	23	27		1	027	0	28	31		Test Ready
24	25	26	27	28	-		010	0	23	28		23 - 000 → AR
28	29	30	31	10		12	016	0	28	27		Test AR
32	33	34	35	16		17	018	0	25	19		Clear L19
36	37	38	39	17	-	42	013	0	28	00		N_t → 00 - 042
40	41	42	43	18	-	20	050	0	00	23		"1" → 23 - 0
44	45	46	47	50		52	052	0	25	28		Clear AR
48	49	50	51	52		53	053	1	00	29		Add L0 → AR
52	53	54	55	53		54	055	3	00	29		Subtract L0 → AR
56	57	58	59	55		57	057	0	28	27		Test AR
60	61	62	63	57	-	60	061	0	23	28		N → AR
64	65	66	67	58	-	55	060	0	00	23		00 - 055 → 23 - 3
68	69	70	71	61	-	20	021	0	00	29		+ "1" → AR
72	73	74	75	21	-	24	031	0	28	23		AR → 23 - 0
76	77	78	79	31	-	42	046	3	00	29		-N_t → AR
80	81	82	83	46		48	049	0	22	31		Test Sign
84	85	86	87	49		50	036	4	17	31		Ring Bell
88	89	90	91	50		52	052	0	25	28		Clear AR
92	93	94	95									
96	97	98	99	13		16	016	0	23	31		Clear (See 16 above)
u0	u1	u2	u3	020		216	000	0	00	10		"1"
u4	u5	u6		042		216	000	0	01	00		N_t

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	36		38	040	0	00	28		x → AR
8	9	10	11	40		42	045	0	28	23		x → 23 - 1
12	13	14	15	45		48	051	1	28	26		x → PN
16	17	18	19	51		53	054	0	28	29		Shift x left
20	21	22	23	54		57	059	1	28	25		2x → ID
24	25	26	27	59		61	062	0	28	29		Shift 2x left
28	29	30	31	62		65	065	1	28	24		4x → MQ
32	33	34	35	65		-68	069	0	23	28		N → AR
36	37	38	39	69		-20	022	3	00	29		-1 → AR
40	41	42	43	22		-24	043	0	28	23		N - 1 → 23 - 0
44	45	46	47	43		45	075	0	28	27		Test AR
48	49	50	51	75		76	077	4	17	31		Ring Bell
52	53	54	55	76		78	078	0	23	28		x → AR
56	57	58	59	78		80	080	0	28	29		2x → AR
60	61	62	63	80		83	083	3	26	29		-PN → AR
64	65	66	67	83		85	085	0	28	27		Test AR
68	69	70	71	85		-89	091	0	23	28		x → AR
72	73	74	75	86		-83	060	0	00	23		00 - 083 → AR
76	77	78	79	91		94	096	0	28	29		4x → AR
80	81	82	83	96		99	099	3	25	29		-ID → AR
84	85	86	87	99		101	104	0	28	27		Test AR
88	89	90	91	104		106	106	0	23	28		x → AR
92	93	94	95	105		-99	060	0	00	23		00 - 099 → AR
96	97	98	99	037		216	00Z	x	v9	70		x
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	106		2	008	0	28	29		8x → AR
8	9	10	11	8		11	011	3	24	29		-MQ → AR
12	13	14	15	11		14	014	0	28	27		Test AR
16	17	18	19	15		-17	019	0	23	28		x → AR
20	21	22	23	19		21	040	0	00	29		+1 → AR
24	25	26	27	40		42	045	0	28	23		
28	29	30	31	45		48	051	1	28	26		
32	33	34	35	51		53	054	0	28	29		
36	37	38	39	54		57	059	1	28	25		
40	41	42	43	59		61	062	0	28	29		
44	45	46	47	59		61	062	0	28	29		
48	49	50	51	62		65	065	1	28	24		See Page 24
52	53	54	55	65		-68	069	0	23	28		
56	57	58	59	69		-20	022	3	00	29		
60	61	62	63	22		-24	043	0	28	23		
64	65	66	67	43		45	075	0	28	27		
68	69	70	71	75		76	077	4	17	31		
72	73	74	75	76		78	078	0	23	28		
76	77	78	79									
80	81	82	83	60		-74	101	0	00	20		"0" → 20 - 2
84	85	86	87	101		1	101	0	28	31		Test Ready
88	89	90	91	102		-2	012	0	00	02-		Format → 02 - 2, 3
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	12		-107	033	0	23	19		23 - 3 → 19 - 107
8	9	10	11	33		35	048	0	91	31		Type 19
12	13	14	15	48		-50	089	0	20	27		Test 20 - 2
16	17	18	19	89		-91	066	0	23	28		23 - 3 → AR
20	21	22	23	90		92	103	0	23	28		23 - 3 → AR
24	25	26	27	66		-82	084	0	00	29		+ Command → AR
28	29	30	31	84		86	105	0	31	31		Obey AR
32	33	34	35	82			000	3	00	2		This command clears AR
36	37	38	39									when proper number is added to it by command 066
40	41	42	43	77		-20	024	1	00	24		"1" → MQ even
44	45	46	47	24		-34	035	0	00	28		y → AR
48	49	50	51	35		38	039	1	28	25		y → ID
52	53	54	55	39		44	044	1	25	20		ID → 20
56	57	58	59	44		49	056	1	25	21		ID → 21
60	61	62	63	56		61	063	1	25	22		ID → 22
64	65	66	67	63		68	068	1	25	23		ID → 23
68	69	70	71	68		-70	100	1	24	28		N → AR
72	73	74	75	100		-20	032	0	00	29		+1 → AR
76	77	78	79	32		-34	038	1	28	24		N + 1 → MQ even
80	81	82	83	38		-42	047	3	00	29		$-N_t$ → AR
84	85	86	87	47		50	072	0	22	31		Test Sign
88	89	90	91	72		73	025	4	17	31		Ring Bell
92	93	94	95	73		75	079	0	00	28		Clear AR
96	97	98	99	034		216	3Zx	u	97	50		y
u0	u1	u2	u3									
u4	u5	u6		103			066	1	25	23		Clear Line 23

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	79		84	088	1	20	29		Add L20
8	9	10	11	88		93	095	3	25	29		-2ID → AR
12	13	14	15	95		97	097	0	28	27		Test AR
16	17	18	19	97		102	001	1	21	29		Add L21
20	21	22	23	98		-95	101	0	00	23		00 - 095 → 23 - 0
24	25	26	27	1		6	007	3	25	29		-2ID → AR
28	29	30	31	7		13	029	0	28	27		Test AR
32	33	34	35	29		34	041	1	22	29		Add L22
36	37	38	39	30		-7	101	0	00	23		00 - 007 → 23 - 3
40	41	42	43	41		46	067	3	25	29		-2ID → AR
44	45	46	47	67		69	070	0	28	27		Test AR
48	49	50	51	70		75	081	1	23	29		Add L23
52	53	54	55	71		-67	101	0	00	23		00 - 067 → 23 - 3
56	57	58	59	81		86	087	3	25	29		-2ID → AR
60	61	62	63	87		89	092	0	28	27		Test AR
64	65	66	67	92		94	094	1	25	29		ID odd → AR
68	69	70	71	93		-87	101	0	00	23		00 - 087 → AR
72	73	74	75	94		-20	035	0	00	29		+ "1" → AR
76	77	78	79	35		38	039	1	28	25		AR → ID
80	81	82	83	39		44	044	1	25	20		
84	85	86	87	44		49	056	1	25	21		See Page 26
88	89	90	91	56		61	063	1	25	22		
92	93	94	95	63		68	068	1	25	23		
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	25		1	025	0	28	31		Test Ready
8	9	10	11	26		28	004	0	15	31		Read Tape
12	13	14	15	4	-	42	005	0	00	28		00 - 042 → AR
16	17	18	19	5		1	005	0	28	31		Test Ready
20	21	22	23	6		7	007	0	19	00		L19 → L0
24	25	26	27	003		216	800	0	00	40		Format
28	29	30	31									
32	33	34	35									
36	37	38	39									
40	41	42	43									
44	45	46	47									
48	49	50	51									
52	53	54	55									
56	57	58	59									
60	61	62	63									
64	65	66	67									
68	69	70	71									
72	73	74	75									
76	77	78	79									
80	81	82	83									
84	85	86	87									
88	89	90	91									
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

TEST ROUTINE NO. 2 - Long Lines Test

Line 0

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	7		30	043	0	28	00		N_t → 00 - 030
8	9	10	11	43		45	045	0	00	28		Command → AR
12	13	14	15	45		47	047	0	28	23		AR → 23 - 2
16	17	18	19	47		49	049	0	31	31		Obey AR
20	21	22	23	44		50	050	0	00	01		L0 → L1, 2, 3, ... or 19
24	25	26	27	50		52	052	3	00	29		-(D = 19) → AR
28	29	30	31	52		54	055	0	22	31		Test Sign
32	33	34	35	55		58	058	4	23	31		Clear
36	37	38	39	56		58	059	0	23	28		23 - 2 → AR
40	41	42	43	59		61	045	0	00	29		+(D = 1) → AR
44	45	46	47									
48	49	50	51	58		60	061	0	00	22		"1" → 22 - 0
52	53	54	55	61		62	063	1	00	30	-	Add L0 → PN
56	57	58	59	63		68	068	1	26	23	-	PN → 23 - 0, 1, 2, 3
60	61	62	63	68		70	070	0	00	28		Command → AR
64	65	66	67	70		71	072	0	28	22		AR → 22 - 3
68	69	70	71	72		76	078	1	23	26	-	23 - 0, 1 → PN
72	73	74	75	78		80	107	0	31	31		Obey AR
76	77	78	79	69		108	028	3	01	30	-	-(L1, 2, 3. . . . , or 19) → PN
80	81	82	83	8		10	012	1	26	27	-	Test PN
84	85	86	87	12		14	015	0	00	29		+(s = 19) → AR
88	89	90	91	13		17	017	2	22	21		22 - 3 → 21 - 0
92	93	94	95	051		050	050	0	00	19		D = 19
96	97	98	99	060		000	000	0	00	01		D = 1
u0	u1	u2	u3	014		216	6w1	w	x3	y0		s = 19
u4	u5	u6		Although the T_n number of 69 is 28, that command is obeyed from AR at 107 time, hence next command is in 8.								

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	15		17	018	0	22	31		Test Sign
8	9	10	11	18		21	021	0	22	28		N → AR
12	13	14	15	19		-23	034	0	22	28		22 - 3 → AR
16	17	18	19	34		36	070	0	00	29		+ (s=1) → AR
20	21	22	23	2								
24	25	26	27	21		-23	024	0	00	29		+1 → AR
28	29	30	31	24		-28	029	0	28	22		N + 1 → 22 - 0
32	33	34	35	29		31	032	3	00	29		$-N_t$ → AR
36	37	38	39	32		35	065	0	22	31		Test Sign
40	41	42	43	65		68	071	4	23	31		Clear
44	45	46	47	66		68	068	0	28	28		Do nothing
48	49	50	51	71		72	087	0	25	19		Clear L19
52	53	54	55	87			000	0	00	19		00 - 088 thru 107 → 19
56	57	58	59			2	074	0	09	31		Type 19
60	61	62	63	74		1	074	0	28	31		Test Ready
64	65	66	67	75		77	083	0	15	31		Read Tape
68	69	70	71	83		-30	084	0	00	28		00 - 030 → AR
72	73	74	75	84		1	084	0	28	31		Test Ready
76	77	78	79	85		86	017	0	19	00		L19 → L00
80	81	82	83	035		000	000	0	01	00		s = 1
84	85	86	87	023		216	000	0	00	10		"1"
88	89	90	91									
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	17		20	020	1	23	26-		23 - 2, 3 → PN
8	9	10	11	20		-24	026	3	23	30-		-(23 - 0, 1) → PN
12	13	14	15	26		29	036	1	26	27		Test PN
16	17	18	19	36		-64	067	0	00	20		See below
20	21	22	23	37		38	039	0	12	31		Gate Type in
24	25	26	27	39		40	041	0	17	31		Ring Bell
28	29	30	31	41		-42	042	0	28	28		Do nothing
32	33	34	35	42		-42	039	0	28	31		Test Ready
36	37	38	39	40		-44	045	0	00	28		Command → AR
40	41	42	43									
44	45	46	47	36		-64	067	0	00	20		Extractor → 20 - 0
48	49	50	51	67		-67	073	2	30	25		20. 21 - 0 → ID even
52	53	54	55	73		10	086	0	26	31		Shift right 5
56	57	58	59	86		-88	001	2	30	28		20. 21 - 0 → AR
60	61	62	63	1		17	022	0	28	29		Shift left 15
64	65	66	67	22		-24	025	0	28	21		AR → 21 - 0
68	69	70	71	25		-27	028	0	00	29		+ Command → AR
72	73	74	75	28		30	030	0	31	31		Obey AR
76	77	78	79	27		118	018	0	00	28-		00 - ? → AR. The timing nos. are altered by comm. 25
80	81	82	83	2		-23	031	0	00	29		+1 → AR
84	85	86	87	31		33	033	2	27	28		20. 21 + 20. AR → AR
88	89	90	91	33		38	038	0	28	20		AR → 20 - 0, 1, 2, 3
92	93	94	95	38		-40	046	0	21	28		21 - 0 → AR
96	97	98	99	064		216	ZZ0	0	00	00		Extractor
u0	u1	u2	u3	003		216	800	0	00	40		Format
u4	u5	u6		023		216	000	0	00	10		"1"

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	46	-	53	054	0	00	29		+ Command → AR
8	9	10	11	54		56	056	0	31	31		Obey AR
12	13	14	15	53		118	025	0	20	00	-	20 - ? → 00 - ?
16	17	18	19	9		11	011	0	00	28		Command → AR
20	21	22	23	11		13	016	1	25	29		+ID even → AR
24	25	26	27	16		18	018	0	31	31		Obey AR
28	29	30	31	10		19	056	0	13	00		L0 → L ? This comm. is modified by com 11
32	33	34	35	57	-	59	012	0	22	28		22 - 3 → AR
36	37	38	39									
40	41	42	43									
44	45	46	47									
48	49	50	51									
52	53	54	55									
56	57	58	59									
60	61	62	63									
64	65	66	67									
68	69	70	71									
72	73	74	75									
76	77	78	79									
80	81	82	83									
84	85	86	87									
88	89	90	91									
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

TEST ROUTINE NO. 2 - Inverting Gates

Line 0

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	17		-86	020	0	28	00		N_t → 00 - 086
8	9	10	11	20		23	025	0	23	31		Clear
12	13	14	15	25		26	000	0	25	19		Clear L19
16	17	18	19			5	005	0	00	21		Constants → 21
20	21	22	23	5		-8	009	0	21	20		"1" → 20 - 0
24	25	26	27	9		14	014	0	21	22		21 → 22
28	29	30	31	14		16	016	0	25	28		Clear AR
32	33	34	35	16		21	027	1	21	29		Add 21
36	37	38	39	27		32	032	3	22	29		Subtract 22
40	41	42	43	32		34	036	0	28	27		Test AR
44	45	46	47	36		-38	039	0	21	28		C_2 → AR
48	49	50	51	37		-32	012	0	00	28		00 - 032 → AR
52	53	54	55									
56	57	58	59	39		41	041	0	20	29		+1 → AR
60	61	62	63	41		43	043	0	28	21		C_2 → 21 - 2
64	65	66	67	43		45	045	0	20	28		C_3 → AR
68	69	70	71	45		-47	049	0	21	29		+1 → AR
72	73	74	75	49		-51	052	0	28	21		C_3 → 21 - 0
76	77	78	79	52		54	054	0	21	28		C_1 → AR
80	81	82	83	54		-56	057	0	20	29		+1 → AR
84	85	86	87	57		-61	062	0	28	21		C_1 → 21 - 0
88	89	90	91	62		-64	065	0	21	28		C_0 → AR
92	93	94	95	001		216	000	0	00	20		Constant C_1
96	97	98	99	002		216	000	0	00	40		Constant C_2
u0	u1	u2	u3	003		216	000	0	00	30		Constant C_3
u4	u5	u6		004		216	000	0	00	10		Constant C_0

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	65		-68	069	0	20	29		+1 → AR
8	9	10	11	69		-72	073	0	28	21		C_0 → 21 - 3
12	13	14	15	73		-86	095	3	00	29		$-N_t$
16	17	18	19	95		07	008	0	22	31		Test Sign
20	21	22	23	8		9	011	4	17	31		Ring Bell
24	25	26	27	9		14	014	0	21	22		21 → 22
28	29	30	31									
32	33	34	35	11		5	007	0	00	21		00 - 1, 2, 3, 4 → 21
36	37	38	39	7		9	019	0	21	20		"1" → 20 - 0
40	41	42	43	19		-21	022	0	21	25		C_1 → ID
44	45	46	47	22		-24	026	2	21	24		C_0 → MQ
48	49	50	51	26		-28	028	0	24	28		$-C_0$ → AR
52	53	54	55	28		30	030	0	22	31		Test Sign
56	57	58	59	30		-28	023	0	00	28		00 - 028 → AR
60	61	62	63	31		34	035	2	21	25-		C_0 → ID
64	65	66	67	35		38	038	2	21	24		C_0 → MQ
68	69	70	71	38		40	042	0	24	28		MQ → AR
72	73	74	75	42		45	046	0	22	31		Test Sign
76	77	78	79	46		-48	050	0	21	25		C_0 → ID
80	81	82	83	47		-43	023	0	00	28		00 - 043 → AR
84	85	86	87									
88	89	90	91									
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	50		-55	056	0	21	24		C_3 → MQ
8	9	10	11	56		58	058	0	24	28		MQ → AR
12	13	14	15	58		60	060	0	22	31		Test Sign
16	17	18	19	60		-58	023	0	00	28		00 - 058 → AR
20	21	22	23	61		-65	066	0	21	25		C_1 → ID
24	25	26	27	66		68	068	0	21	24		C_3 → MQ
28	29	30	31	68		70	070	0	24	28		MQ → AR
32	33	34	35	70		72	076	0	22	31		Test Sign
36	37	38	39	76		-80	081	0	20	28		N → AR
40	41	42	43	77		-70	023	0	00	28		00 - 070 → AR
44	45	46	47									
48	49	50	51	81		83	083	0	00	29		"1" → AR
52	53	54	55	83		85	085	0	28	20		AR → 20 - 0
56	57	58	59	85		87	087	3	00	29		$-N_t$ → AR
60	61	62	63	87		89	018	0	22	31		Test Sign
64	65	66	67	18		19	071	4	17	31		Ring Bell
68	69	70	71	71		1	071	0	28	31		Test Ready
72	73	74	75	72		74	088	0	15	31		Read Tape
76	77	78	79	88		1	088	0	28	31		Test Ready
80	81	82	83	89		90	090	1	19	29		Add L19
84	85	86	87	90		92	092	0	28	27		Test AR
88	89	90	91	92		-86	099	0	00	28		N_t → AR
92	93	94	95	93		95	096	0	08	31		Type AR
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	96		1	096	0	28	31		Test Ready
8	9	10	11	97		-90	094	0	00	28		00 - 090 → AR
12	13	14	15	94		96	091	0	08	31		Type AR
16	17	18	19	91		1	091	0	28	31		Test Ready
20	21	22	23									
24	25	26	27	99		100	017	0	19	00		L19 → L00
28	29	30	31									
32	33	34	35	12		1	012	0	28	31		Test Ready
36	37	38	39	13		-107	015	0	28	19		AR → 19 - 107
40	41	42	43	15		17	036	0	09	31		Type 19
44	45	46	47									
48	49	50	51	23		1	023	0	28	31		Test Ready
52	53	54	55	24		-107	029	0	28	19		AR → 19 - 107
56	57	58	59	29		31	076	0	09	31		Type 19
60	61	62	63									
64	65	66	67									
68	69	70	71									
72	73	74	75									
76	77	78	79									
80	81	82	83									
84	85	86	87									
88	89	90	91									
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

TEST ROUTINE NO. 2 - Overflow & End-Around Line 0

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	17		-86	022	0	28	00		N_t → 00 - 086
8	9	10	11	22		25	025	0	23	31		Clear
12	13	14	15	25		30	030	0	25	20		Clear 20
16	17	18	19	30		-32	034	2	00	21-		"1" → 21 - 1
20	21	22	23	34		36	036	0	29	31		Test Overflow
24	25	26	27	36		-41	042	0	00	20		A → 20 - 1
28	29	30	31	37		-41	042	0	00	20		A → 20 - 1
32	33	34	35	42		44	044	0	00	20		B → 20 - 3
36	37	38	39	44		46	046	1	20	28		A → AR
40	41	42	43	46		48	048	1	20	29		+B → AR
44	45	46	47	48		50	050	0	29	31		Test Overflow
48	49	50	51	50		-48	098	0	00	28		00 - 048 → AR
52	53	54	55	51		-53	054	3	20	28		(-A) → AR
56	57	58	59	54		56	056	3	20	29		-B → AR
60	61	62	63	56		58	058	0	29	31		Test Overflow
64	65	66	67	58		-56	098	0	00	28		00 - 056 → AR
68	69	70	71	59		-61	062	3	20	28		(-A) → AR
72	73	74	75	62		-65	066	3	20	29		-A → AR
76	77	78	79	66		68	068	0	29	31		Test Overflow
80	81	82	83	68		-66	098	0	00	28		00 - 066 → AR
84	85	86	87	69		-73	074	1	00	28		x → AR
88	89	90	91	041		216	800	0	00	00		A
92	93	94	95	043		216	800	0	00	10		B
96	97	98	99	032		216	000	0	00	10		"1"
u0	u1	u2	u3	073		See	Page	3				x
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	74		76	076	3	20	29		-B → AR
8	9	10	11	76		78	078	0	29	31		Test Overflow
12	13	14	15	78		-80	082	1	20	26-		A → PN
16	17	18	19	79		-76	096	0	00	28		00 - 076 → AR
20	21	22	23									
24	25	26	27	82		-86	088	3	20	30-		-B → PN
28	29	30	31	88		90	090	0	29	31		Test Overflow
32	33	34	35	90		-92	094	1	20	26		A → PN
36	37	38	39	91		-88	096	0	00	28		00 - 088 AR
40	41	42	43									
44	45	46	47	94		-98	100	1	20	30		+B → PN
48	49	50	51	100		102	102	0	29	31		Test Overflow
52	53	54	55	102		-100	098	0	00	28		00 - 100 → AR
56	57	58	59	103		106	106	0	23	31		Clear
60	61	62	63	106		-	004	3	20	30		(-A) → PN
64	65	66	67	4		-6	008	3	20	30		-B → PN
68	69	70	71	8		10	010	0	29	31		Test Overflow
72	73	74	75	10		-8	098	0	00	28		00 - 008 → AR
76	77	78	79	11		14	014	0	23	31		Clear
80	81	82	83	14		-16	018	3	20	31		(-A) → PN
84	85	86	87	18		-20	023	3	20	30		-A → PN
88	89	90	91	23		25	026	0	29	31		Test Overflow
92	93	94	95	26		-23	098	0	00	28		00 - 023 → AR
96	97	98	99	27		-31	033	3	20	28		(-A) → AR
u0	u1	u2	u3	33		34	035	3	25	29		-0 → AR
u4	u5	u6		35		37	038	0	22	31		Test Sign

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	39		42	045	1	20	26-		B → PN
8	9	10	11	45		48	049	3	25	30-		-0 → PN
12	13	14	15	49		51	052	1	26	28		PN even → AR
16	17	18	19	52		54	060	0	22	31		Test Sign
20	21	22	23	60		-65	067	0	21	28		N → AR
24	25	26	27	61		-52	098	0	00	28		00 - 052 → AR
28	29	30	31									
32	33	34	35	67		-72	073	0	00	29		+1 → AR
36	37	38	39	73		-77	085	0	28	21		N + 1 → 21 - 1
40	41	42	43	85		87	085	3	00	29		$-N_t$ → AR
44	45	46	47	87		89	104	0	22	31		Test Sign
48	49	50	51	104		105	006	4	17	31		Ring Bell
52	53	54	55	105		107	044	0	28	28		Do nothing
56	57	58	59									
60	61	62	63	6		86	092		00	28		N_t → AR
64	65	66	67	92		1	092	0	28	31		Test Ready
68	69	70	71	93		95	070	0	15	31		Read Tape
72	73	74	75	70		1	070	0	28	31		Test Ready
76	77	78	79	71		72	017	0	19	00		L19 → L0
80	81	82	83									
84	85	86	87	38		-35	098	0	00	28		00 - 035 → AR
88	89	90	91									
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	96		98	098	3	00	29		$(T_n = 1) \rightarrow$ AR
8	9	10	11	98		1	098	0	28	31		Test Ready
12	13	14	15	99		100	101	0	25	19		Clear L19
16	17	18	19	101		-10	005	0	28	19		AR → 19 - 107
20	21	22	23	5			012	0	09	31		Type 19
24	25	26	27	12		-16	019	0	28	21		AR → 21 - 0
28	29	30	31	19		21	021	0	00	20		Extractor → 20 - 0
32	33	34	35	21		-24	028	0	31	28		20. 21 - 0 → AR
36	37	38	39	28		-32	040	0	25	20		Clear 20 - 0
40	41	42	43	40		-47	053	0	00	29		+ Command → AR
44	45	46	47	53		55	055	0	31	31		Obey AR
48	49	50	51	47		58	001	0	23	31		Clear and return
52	53	54	55									to proper place in
56	57	58	59									routine. The T_n
60	61	62	63									number of 047 is
64	65	66	67									altered.
68	69	70	71	020		216	007	Z	00	00		Extractor
72	73	74	75	097		000	001	0	00	00		T_n = 1
76	77	78	79									
80	81	82	83									
84	85	86	87									
88	89	90	91									
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

TEST ROUTINE NO. 2 - Multi. -Div., Shift - Norm. Logic Line 0

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	N O T E S
4	5	6	7	17		-19	020	0	28	22		N_t → 22 - 3
8	9	10	11	20		-25	026	0	00	28		"1" → AR
12	13	14	15	26		-28	031	0	28	22-		"1" → 22 - 0, 1
16	17	18	19	31		-34	035	0	00	22		Exit 1 → 22 - 2
20	21	22	23	35		38	043	0	00	21		A → 21 - 0; B → 21 - 1
24	25	26	27	43		46	046	2	21	25-		A → I D odd
28	29	30	31	46		-49	051	0	21	24		B → MQ
32	33	34	35	51		56	000	0	24	31		Mult.
36	37	38	39			-4	006	0	26	23		A. B → 23 - 0, 1
40	41	42	43	6		-8	013	2	21	25		A → ID odd
44	45	46	47	13		-16	0 9	0	23	26		A. B → PN
48	49	50	51	19		58	078	1	25	31		Divide
52	53	54	55	78		-81	082	1	21	28		B → AR
56	57	58	59	82		-84	085	0	24	20		$\frac{A.B}{A}$ → 20 - 0
60	61	62	63	85		-88	089	3	20	29		$-\frac{A.B}{A}$ → AR
64	65	66	67	89		91	091	0	28	27		Test AR
68	69	70	71	91		-93	094	0	22	28		"1" → AR
72	73	74	75	92		95	095	0	23	31		Clear
76	77	78	79	025		216	000	0	00	10		"1"
80	81	82	83	036		216	987		54	30		A
84	85	86	87	037		216	345	6	78	90	-	B
88	89	90	91									
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	94		-96	097	0	22	29		+N → AR
8	9	10	11	97		-100	101	0	28	22		N + 1 → 22 - 0
12	13	14	15	101		-103	104	3	22	29		$-N_t$ → AR
16	17	18	19	104		106	004	0	22	31		Test Sign
20	21	22	23	4		5	022	4	17	31		Ring Bell
24	25	26	27	5		7	008	0	22	28		Command 1 → AR
28	29	30	31	8		10	010	0	31	31		Obey AR
32	33	34	35						EXIT I			
36	37	38	39	34		-12	014	1	21	28		A → AR
40	41	42	43	14		16	016	1	00	29		+ A → AR
44	45	46	47	16		20	021	1	28	21		A + A → 21 - 0
48	49	50	51	21		25	027	1	21	28		B → AR
52	53	54	55	2		29	029	1	00	29		B → AR
56	57	58	59	29		33	043	1	28	21		B + B → 21 - 1
60	61	62	63									
64	65	66	67	95		97	098	0	21	25		B → ID
68	69	70	71	98		100	103	2	21	24		A → MQ
72	73	74	75	103		58	054	0	24	31		Mult.
76	77	78	79	54		-56	057	0	26	20-		B=A → 20 - 0, 1
80	81	82	83	57		-60	062	1	20	26		B. A → PN
84	85	86	87	62		-64	066	3	23	30		- AB → PN
88	89	90	91	66		-68	070	1	26	27		Test PN
92	93	94	95	70		72	074	0	00	28		Exit 2 → AR
96	97	98	99	71		-73	074	0	00	28		Exit 3 → AR
u0	u1	u2	u3	015		216	001	2	34	50		A
u4	u5	u6		028		216	000	Z		wx	y0	B

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	74		1	074	0	28	31		Test Ready
8	9	10	11	75		78	079	0	23	31		Clear
12	13	14	15	79		80	080	0	25	19		Clear L19
16	17	18	19	80		-107	001	0	28	19		AR → 19 - 107
20	21	22	23	1		4	007	0	00	02		Format → 2
24	25	26	27	7		10	011	0	09	31		Type
28	29	30	31	11		13	013	0	31	31		Obey AR
32	33	34	35									
36	37	38	39	72		15	091	0	20	20		Exit 2
40	41	42	43	73		16	091	0	21	21		Exit 3
44	45	46	47									
48	49	50	51	22		25	030	0	22	28		"1" AR
52	53	54	55	30		32	039	0	28	22		"1" → 22 - 0
56	57	58	59	39		42	042	0	23	31		Clear
60	61	62	63	42		-44	045	0	21	24		A → MQ
64	65	66	67	45		48	053	1	25	28		Clear AR
68	69	70	71	53		100	009	0	27	31		Normalize
72	73	74	75	9		12	012	0	24	25		MQ → ID
76	77	78	79	12		14	018	0	24	20		MQ → 20 - 1
80	81	82	83	18		20	023	3	28	28		(-AR) → AR
84	85	86	87	23		100	083	0	26	31		Shift
88	89	90	91	83		85	086	0	24	27		Test MQ
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	86		-88	090	0	25	28		ID → AR
8	9	10	11	90		-92	093	3	21	29		-A → AR
12	13	14	15	93		95	099	0	28	27		Test AR
16	17	18	19	99		101	102	0	22	28		N → AR
20	21	22	23	100		-99	074	0	00	28		00 - 099 → AR
24	25	26	27									
28	29	30	31	102		-105	106	0	22	29		+1 → AR
32	33	34	35	106			010	0	28	22		N + 1 → 22 - 0
36	37	38	39	10		12	024	3	22	29		- N_t → AR
40	41	42	43	24		26	038	0	22	31		Test Sign
44	45	46	47	38		39	040	4	17	31		Ring Bell
48	49	50	51	39		42	042	0	23	31		Clear
52	53	54	55									
56	57	58	59	87		89	074	0	00	28		00 - 088 → AR
60	61	62	63									
64	65	66	67	88		20	099	0	23	31		Clear
68	69	70	71									
72	73	74	75	60		1	060	0	28	31		Test Ready
76	77	78	79	61		63	064	0	15	31		Read Tape
80	81	82	83	64		1	064	0	28	31		Test Ready
84	85	86	87	65		-67	068	0	22	28		N_t → AR
88	89	90	91	68		69	017	0	19	00		L19 → L0
92	93	94	95									
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	40		42	044	0	22	28		"1" → AR
8	9	10	11	44	-	48	049	0	28	22		"1" → 22 - 0
12	13	14	15	49	-	84	055	0	00	20		C → 20 - 0
16	17	18	19	55		57	059	0	00	21		D → 21 - 0
20	21	22	23	59		61	063	0	31	28		C· D → AR
24	25	26	27	63		65	067	0	30	29		+ (C· D) → AR
28	29	30	31	67		69	069	3	21	29		- D → AR
32	33	34	35	69		71	076	0	28	27		Test AR
36	37	38	39	76	-	80	081	0	21	28		D → AR
40	41	42	43	77	-	76	074	0	00	28		00 - 076 → AR
44	45	46	47									
48	49	50	51	81	-	84	096	0	27	28		+ C· D + C (C+D) → AR
52	53	54	55	96	-	100	105	3	21	29		- C → AR
56	57	58	59	105		105	032	0	28	27		Test AR
60	61	62	63	32		36	041	0	22	28		N → AR
64	65	66	67	33		32	074	0	00	28		00 - 032 → AR
68	69	70	71									
72	73	74	75	41	-	45	047	0	22	29		+ 1 → AR
76	77	78	79	47		49	050	0	28	22		N + 1 → 22 - 0
80	81	82	83	50		52	052	3	22	29		$-N_t$ → AR
84	85	86	87	52		54	058	0	28	27		Test AR
88	89	90	91	58		59	060	4	17	31		Ring Bell
92	93	94	95	59		61	063	0	31	28		See 059 Above
96	97	98	99	084		216	999	9	99	90		C
u0	u1	u2	u3	056		216	3w3	w	3w	30		D
u4	u5	u6										

TEST ROUTINE NO. 2 - Type-out, Computation Test

Line 0

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	17		19	044	0	15	31		Read Tape
8	9	10	11	44		1	044	0	28	31		Test Ready
12	13	14	15	45		46	018	0	19	06		L19 → L6
16	17	18	19	18		21	032	0	23	31		Clear
20	21	22	23	32		34	034	0	15	31		Read Tape
24	25	26	27	34		1	034	0	28	31		Test Ready
28	29	30	31	35		36	043	0	19	04		L19 → L4
32	33	34	35	43		44	046	0	25	19		Clear 19
36	37	38	39	46		-86	087	0	28	00		N_t → 00 - 086
40	41	42	43	87		92	101	0	00	20		Not useful
44	45	46	47	101		106	106	0	00	19		00 - 102, 3, 4, 5 → 19
48	49	50	51	106		4	007	0	00	02		Format → 02
52	53	54	55	7		-54	056	0	00	20-		00 - 054, 5 → 20 - 2, 3
56	57	58	59	56		-106	006	0	20	19-		20 - 2, 3 → 19 - 106, 7
60	61	62	63	6		8	008	0	09	31		Type 19
64	65	66	67	8		12	024	0	00	22		S_0 → 22 - 1; "1" → 22 - 2 0 → 22 - 3
68	69	70	71	24		27	027	2	22	20		0 → 20 - 2
72	73	74	75	27		29	029	0	00	28		Exit 1 → AR
76	77	78	79	29		30	085	0	06	01		Line 06 → -L1
80	81	82	83	85		87	093	5	21	31		Commands from L 1
84	85	86	87	003		216	800	0	00	x0		
88	89	90	91	002		216	000	0	0w	60		Format
92	93	94	95	001		216	000	0	0y	80		
96	97	98	99	009		216	345	6	78	90		0_0
u0	u1	u2	u3	010		216	000	0	00	10		"1"
u4	u5	u6		011		216	002	3	45	60		0

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7			100	100	4	21	31		Exit 1; Comm. from L0
8	9	10	11	100	-	104	004	0	20	22		Sin 0 → 22 - 0
12	13	14	15	4	-	36	037	0	00	28		Exit 2 → AR
16	17	18	19	37	-	057	078	1	21	31		Commands from L1
20	21	22	23			56	057	4	20	31		Exit 2; Comm. from L0
24	25	26	27	57		59	059	1	20	28		ϕ = Arcsin (sin 0) → AR
28	29	30	31	59		61	062	3	22	29		-0 → AR
32	33	34	35	62		64	064	1	28	20		ϕ - 0 → 20 - 3
36	37	38	39	64	-	67	068	0	00	21		Extractor → 21 - 3
40	41	42	43	68	-	71	072	0	31	27		20· 21 - 3 → Test
44	45	46	47	72		74	074	1	22	28		0 → AR
48	49	50	51	73	-	68	080	0	00	28		00 - 068 → AR
52	53	54	55									
56	57	58	59	74		76	077	3	22	29		- 0 → AR
60	61	62	63	77	-	81	082	1	28	22		0 - 0 → 22 -1
64	65	66	67	82	-	86	088	0	22	28		N → AR
68	69	70	71	88	-	14	015	0	00	29		+1 → AR
72	73	74	75	15	-	18	019	0	28	22		N + 1 → 22 - 2
76	77	78	79	19	-	86	092	3	00	29		$-N_t$ → AR
80	81	82	83	92		94	023	0	22	31		Test Sign
84	85	86	87	23	-	25	030	0	00	28		"7" → AR
88	89	90	91	067		216	ZZZ	Z	ZZ	y0		Extractor
92	93	94	95	014		216	000	0	00	10		"1"
96	97	98	99	025		216	000	0	00	70		"7"
u0	u1	u2	u3									
u4	u5	u6										

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	30		1	030	0	28	31		Test Ready
8	9	10	11	31		33	076	0	06	31		Reverse Tape Search
12	13	14	15	76	-	14	020	3	00	29		-"1" → AR
16	17	18	19	20		1	020	0	28	31		Test Ready
20	21	22	23	21		23	038	0	22	31		Test Sign
24	25	26	27	38		40	076	0	06	31		Reverse Tape Search
28	29	30	31	39		35	040	0	15	31		Read Tape
32	33	34	35	40		1	040	0	28	31		Test Ready
36	37	38	39	41		43	047	0	28	28		Do nothing
40	41	42	43	47	-	86	093	0	00	28		N_t → AR
44	45	46	47	93		1	093	0	28	31		Test Ready
48	49	50	51	93		1	093	0	28	31		(This is a repeat)
52	53	54	55	94		95	017	0	19	00		L19 → L0
56	57	58	59									
60	61	62	63	80		1	080	0	28	31		Test Ready
64	65	66	67	81	-	107	052	0	28	19		AR → 19 - 107
68	69	70	71	52		54	072	0	09	31		Type 19
72	73	74	75	055		216	112	2	33	40	-	
76	77	78	79	054		216	445	5	66	70		
80	81	82	83	105		216	778	8	99	00		Test Type Out
84	85	86	87	104		216	uuv	v	ww	x0	-	numbers
88	89	90	91	103		216	xxy	y	ZZ	00		
92	93	94	95	102		216	002	3	45	00		
96	97	98	99									
u0	u1	u2	u3									
u4	u5	u6										

P.P.R. LANGUAGE

Commands are ordinarily written by programmers in decimal notation rather than in hexadecimal as shown on drawing 5. A routine, such as the PROGRAM PREPARATION ROUTINE (P.P.R.), will convert the decimally coded command into binary - using the G-15 to do the work. Decimal notation of a command in P.P.R. language is as follows:

L (p) T N C S D (s)

L - COMMAND LOCATION (2 digits: 00 - u7*).

(p) - PREFIX - this is not required except as follows:

a command with DESTINATION ≠ 31 will be DEFERRED unless a prefix of "u" (for "urgent") makes it IMMEDIATE.

a command with DESTINATION = 31 will be IMMEDIATE unless a prefix of "w" (for "wait") makes it DEFERRED.

T - TIMING NUMBER (2 digits: 00 - w7*).

N - NEXT COMMAND LOCATION (2 digits: 00 - u7*).

C - "C" CODE (1 digit : 0 - 7) - "C" CODES define both the CHARACTERISTIC and the SINGLE/DOUBLE PRECISION bit by considering the S/D bit as a 2^2 bit applied the left of the CH bits, hence:

C = 0 - 3 means: CH = 0 - 3, SINGLE PRECISION
C = 4 - 7 means: CH = 0 - 3, DOUBLE PRECISION

(i.e. add 4 to the CHARACTERISTIC to denote DOUBLE PRECISION)

S - SOURCE NUMBER (2 digits: 00 - 31).

D - DESTINATION NUMBER (2 digits: 00 - 31).

(s) - SUFFIX - A MINUS SIGN is applied as a suffix if a BREAK POINT is desired in a command.

EXAMPLE: a command coded as u76 u5 5 13 26- is equivalent to:
I/D = 0, T = 76, BP = 1, N = 105, CH = 1, S = 13, D = 26, S/D = 1.

*NOTE: L, T, and N numbers higher than 99 are expressed by using 10 = u, 11 = v, and 12 = w in the 10^0 digit position (e.g. 106 = u6)

FUNDAMENTAL OPERATIONS - BOOLEAN ALGEBRA

The system of notation employed for describing the operation of the computer is variously called BOOLEAN ALGEBRA, the PROPOSITIONAL CALCULUS, or LOGICAL ALGEBRA. The basic symbols are letters that, in this case, are interpreted as signals which may have two values, HIGH and LOW. The basic connectives are "+", ".", and "$\overline{\ }$",which approximate the usage of "or", "and", and "not" respectively, in ordinary language. The following tables, called TRUTH TABLES, explain the meaning attached to the connectives, where "1" indicates a high signal and "0" a low signal.

		"OR"	"AND"	"NOT"
A	B	A+B	A·B	$\overline{A}$
1	1	1	1	0
1	0	1	0	0
0	1	1	0	1
0	0	0	0	1

In writing expressions, the following convention may be adopted to reduce the use of parenthesis: "·" binds two symbols more strongly than "+". This, it will be seen, is the same convention adopted in the normal numerical algebra if "·" is treated as "x" (times) and "+" is treated as "+" (plus). For example, the algebraic expression (a x b) + (c x d) is written as a x b + c x d; similarly, the logical expression (A · B) + (C · D) is written as A · B + C · D. Again, $\left[a + (b \times c)\right]$ x d is written (a + b x c) x d and $\left[A + (B \cdot C)\right]$ · D is written (A + B · C) · D.

Using the TRUTH TABLE given above, conditions under which a given combination of signals is high may be derived. For example, given the combination of signals

$$D = A \cdot B + (A+C) \cdot (A+\overline{B}),$$

the following table lists the steps employed in deriving the conditions of A, B, and C under which D is high. Note that the first step is to write all possible combinations of the given terms, and that the number of these combinations is 2^n, where n is the number of different letters in the expression.

A B C	$A\cdot B$	$A+C$	$A+\overline{B}$	$(A+C)\cdot(A+\overline{B})$	$A\cdot B+(A+C)\cdot(A+\overline{B})$
1 1 1	1	1	1	1	1
1 1 0	1	1	1	1	1
1 0 1	0	1	1	1	1
1 0 0	0	1	1	1	1
0 1 1	0	1	0	0	0
0 1 0	0	0	0	0	0
0 0 1	0	1	1	1	1
0 0 0	0	0	1	0	0

In addition, to determine the conditions under which a given combination of signals is high, it is also desirable that the expression representing a given configuration of signals be in its simplest form. "Simplest" in this case is difficult to define in abstract generality; however, in any given instance the relative simplicity of two forms of a given expression is not difficult to determine. In order to manipulate logical expressions, the following rules are employed:

Associativity	$(A \cdot B)\cdot C = A \cdot(B \cdot C) = A \cdot B \cdot C$	(a)	I
Distributivity	$A \cdot(B + C) = A \cdot B + A \cdot C$	(b)	
Commutivity	$A \cdot B = B \cdot A$	(a)	II
	$A + B = B + A$	(b)	
DeMorgan's Law	$\overline{A + B} = \overline{A} \cdot \overline{B}$	(a)	III
	$\overline{A \cdot B} = \overline{A} + \overline{B}$	(b)	

$$\left.\begin{array}{ll} B \cdot (A + \bar{A}) = B & (a) \\ B + A \cdot \bar{A} = B & (b) \end{array}\right\} \text{IV}$$

$$\left.\begin{array}{ll} A + \bar{A} \cdot B = A + B & (a) \\ A + A \cdot B = A & (b) \end{array}\right\} \text{V}$$

$$\left.\begin{array}{ll} A \cdot A = A & (a) \\ A + A = A & (b) \end{array}\right\} \text{VI}$$

$$\bar{\bar{A}} = A \qquad \text{VII}$$

Consider now the expression: $C \cdot \bar{B} \cdot \bar{A} + C \cdot \bar{B} \cdot A + \overline{\bar{C} \cdot B \cdot A + \bar{C} \cdot B \cdot \bar{A}}$

Applying rules

IIIa $C \cdot \bar{B} \cdot \bar{A} + C \cdot \bar{B} \cdot A + \overline{\bar{C} \cdot B \cdot A} \cdot \overline{\bar{C} \cdot B \cdot \bar{A}}$

IIIb $C \cdot \bar{B} \cdot \bar{A} + C \cdot \bar{B} \cdot A + (\bar{\bar{C}} + \bar{B} + \bar{A}) \cdot (\bar{\bar{C}} + \bar{B} + \bar{\bar{A}})$

VII $C \cdot \bar{B} \cdot \bar{A} + C \cdot \bar{B} \cdot A + (C + \bar{B} + \bar{A}) \cdot (C + \bar{B} + A)$

Ib $C \cdot \bar{B} \cdot \bar{A} + C \cdot \bar{B} \cdot A + C \cdot C + C \cdot \bar{B} + C \cdot A + \bar{B} \cdot C + \bar{B} \cdot \bar{B} + \bar{B} \cdot A + \bar{A} \cdot C + \bar{A} \cdot \bar{B} + \bar{A} \cdot A$

VIa,b $C \cdot \bar{B} \cdot \bar{A} + C \cdot \bar{B} \cdot A + C + C \cdot \bar{B} + C \cdot A + \bar{B} + \bar{B} \cdot A + C \cdot \bar{A} + \bar{B} \cdot \bar{A} + \bar{A} \cdot A$

IVb $C \cdot \bar{B} \cdot \bar{A} + C \cdot \bar{B} \cdot A + C + C \cdot \bar{B} + C \cdot A + \bar{B} + \bar{B} \cdot A + C \cdot \bar{A} + \bar{B} \cdot \bar{A}$

II $C \cdot \bar{B} \cdot \bar{A} + C \cdot \bar{B} \cdot A + C + C \cdot \bar{B} + \bar{B} \cdot \bar{A} + \bar{B} \cdot A + C \cdot A + C \cdot \bar{A} + \bar{B}$

Ib $C \cdot \bar{B} \cdot (\bar{A} + A) + C + C \cdot \bar{B} + \bar{B} \cdot (\bar{A} + A) + C \cdot (A + \bar{A}) + \bar{B}$

IVa $C \cdot \bar{B} + C + C \cdot \bar{B} + \bar{B} + C + \bar{B}$

VIb $C \cdot \bar{B} + C + \bar{B}$

V $C + \bar{B}$

therefore $C \cdot \bar{B} \cdot \bar{A} + C \cdot \bar{B} \cdot A + \overline{\bar{C} \cdot B \cdot A + \bar{C} \cdot B \cdot \bar{A}} = C + \bar{B}$

FUNDAMENTAL OPERATIONS - BINARY ARITHMETIC

EXAMPLE OF BINARY NOTATION

$118\frac{9}{16}$ is a quantity expressed in decimal notation. It may also be expressed in terms of the following:

$$\begin{aligned}118\frac{9}{16} &= 64 + 32 + 16 + 4 \;\; + 2 \;\;\; + 1/2 + 1/16\\ &= 2^{6} + 2^{5} + 2^{4} + 2^{2} + 2^{1} \;\; + 2^{-1} + 2^{-4}\end{aligned}$$

The presence or absence of 2^{i} in the quantity above may be expressed in terms of "0" and "1" multipliers as shown below. The BINARY POINT indicates the location of 2^{0}. (This is a binary number.)

$$\begin{array}{ccccccccccccccccc} 2^{n}\cdots\cdot 2^{8} & +2^{7} & +2^{6} & +2^{5} & +2^{4} & +2^{3} & +2^{2} & +2^{1} & +2^{0} & & +2^{-1} & +2^{-2} & +2^{-3} & +2^{-4} & +2^{-5} & +2^{-6} & \cdots\cdot 2^{-n}\\ o\cdots\cdot o & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & . & 1 & 0 & 0 & 1 & 0 & & o\cdots\cdot o\\ & & & & & & & & & \blacktriangle & & & & & & & \end{array}$$

This may also be expressed in terms of an integral number:

$$\begin{array}{cccccccccccccc} (& 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & .) \times 2^{-4}\\ & & & & & & & & & & & & & \blacktriangle \end{array}$$

- or a fractional number:

$$\begin{array}{cccccccccccccc} (. & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 &) \times 2^{8}\\ \blacktriangle & & & & & & & & & & & & & \end{array}$$

- or in terms of groups of four binary digits:

$$(0111.) \times 2^{4} \qquad (0110.) \times 2^{0} \qquad (1001.) \times 2^{-4}$$

- which may be represented by hexadecimal digits (0 - z):

$$7 \times 16^{1} \qquad\qquad 6 \times 16^{0} \qquad\qquad 9 \times 16^{-1}$$

- abbreviated: 76.9_{16}

- conclusion: decimal $118\frac{9}{16}$ = binary 1110110.1001 = hexadecimal 76.9

CONVERSION OF A BINARY NUMBER TO DECIMAL

$$\begin{aligned}1110110.1001 &= 2^{6} + 2^{5} + 2^{4} + 2^{2} + 2^{1} + 2^{-1} + 2^{-4}\\ &= 64 + 32 + 16 + 4 \;\; + 2 \;\; + 1/2 + 1/16 \;\; = \;\; 118\frac{9}{16}\end{aligned}$$

CONVERSION OF A DECIMAL NUMBER TO BINARY

```
118 9/16 = ?     118 9/16
                 -64          2^6
                  54 9/16
                 -32          2^5
                  22 9/16
                 -16          2^4
                   6 9/16
                 - 4          2^2
                   2 9/16
                 - 2          2^1
                     9/16
                 -   8/16     2^-1
                     1/16
                 -   1/16     2^-4
                   ----
```

```
   2^6  2^5  2^4       2^2  2^1        2^-1           2^-4

 = 1    1    1    0    1    1    0  .  1    0    0    1
                                    ▲
```

ADDITION IN BINARY

		INPUT	OUTPUT	
AUGEND	ADDEND	CARRY	CARRY	SUM
U	D	C	C_s	A
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

EXAMPLE:

```
CAR.   1   1 1
AUG.     1 0 1 1 0.  = 22
ADD.     1 0 1 1 1.  = 23
SUM.   1 0 1 1 0 1.  = 45
```

G-15D NUMBERS (28 binary digits and /sign):

EXAMPLE is decimal $118\frac{9}{16}$ =

.0111011010010000000000000000/0 x 2^8 =(.7690000$_{16}$) x 2^8

or

.1110110100100000000000000000/0 x 2^7 =(.yx20000$_{16}$) x 2^7 (NORMALIZED)

or

.0000000000000000011101101001/0.x 2^{24}=(.0000769$_{16}$) x 2^{24} (lowest order bit = 1/16)

or

.1111111111111111100010010111/1 x 2^{24}=(.zzzz897$_{16}$) x 2^{24} (COMPLEMENTED)

INDEX OF FREQUENTLY-REFERRED-TO SIGNALS

SIGNAL	ORIGIN	DESCRIPTION AND (REFERENCE DRAWING)
①	G	CHAR. = 1 (23)
②	G	CHAR. = 2 (23)
③	G	CHAR. = 3 (23)
④	G	CHAR. = 0 (23)
AC	FF	ACCUMULATOR REG. CARRY FF (27)
AR	FF	ACCUMULATOR REG. READ FF; also abbrev. for reg. (27)
*AS	FF	AUTOMATIC/STANDARD FF (in AN models only) (67, 68)
AUTO	G	Enable for AUTOMATIC RELOAD circuit in the AN INPUT SYSTEM (= AS·SLOW-IN) (67)
BP		abbrev. for BREAK POINT (5, 30)
*C1 *C2-C6 *C7-CV *CX,CW	FF FF's FF's FF's	S/D STATIC FF DESTINATION STATIC FF's SOURCE STATIC FF's CHARACTERISTIC STATIC FF's (30)
CC	FF	COMMAND REG. CARRY FF (22, 27)
*CD's	FF's	COMMAND LINE designators (22)
CE $\overline{CE}$	FF FF	High during EVEN word times High during ODD word times (15)
*CG	FF	"NEXT COMMAND FROM AR" control FF (22)
*CH CZ	FF FF	START-STOP control FF's (30)
CI	BI	COMMAND info. (not inverted) (22)
CJ	FF	FF controlling READ COMMAND signal (30)
CM	FF	COMMAND REG. READ FF; also abbrev. for reg. (22, 27)
CN	FF	NUMBER TRACK READ FF; also abbrev. for track (15, 57)
*CQ	FF	TEST FF; when set, next comm. is read from WT = N+1 (30)
CS	BI	"VIA AR" CHARACTERISTIC (26)

* (neon indicator)

D0-D7 DU-DX	G's G's	DESTINATION designators (23)
(DA-1		DIGITAL DIFFERENTIAL ANALYZER attachment)
DS	G	TRANSFER when DEST. is 31 (= TR·D7·DX) (23)
Ⓔ	G	selected TF pulse initiating INPUT CODE process (48, 68)
EB	G	abbrev. for EARLY BUS (26, 28)
*FO	FF	OVERFLOW FF (27)
HC	G	sync. signal from INPUT/OUTPUT media (47, 48, 52, 68, 69)
IB	G	abbrev. for INTERMEDIATE BUS (26)
IC	FF	"when to invert" FF in INVERTING GATES (26)
ID		abbrev. for MULTIPLICAND-DENOMINATOR REG. (PJ is READ FF) (34, 36, 40)
*IP	FF	SIGN FF in INV. GATES (for two-word lines) (26, 34, 40)
(IR		abbrev. for INPUT-OUTPUT REG. attachment output)
IS	FF	"whether to complement" FF in INV. GATES (26)
"L"		word-time LOCATION of a COMMAND (5, 20)
LB	G	abbrev. for LATE BUS (26)
M0-M23	FF's	READING FF's of LINES 00-23 (also used as abbrev. for lines) (21)
MQ		abbrev. for MULTIPLIER-QUOTIENT REG. (PR is READ FF) (34, 36, 40)
MZ	FF	LINE Z READ FF; also abbrev. for line (48, 52, 60, 68, 69)
"N"		COMMAND info.: location of NEXT COMMAND (5, 20)
OA's	FF's	4-bit reg. used in INPUT/OUTPUT precession activities (48, 52, 60, 68, 69)
OB's	FF's	5 FF's in INPUT/OUTPUT system - sometimes used as 5-bit reg. (48, 52, 60, 68, 69)
*OC's	FF's	4-bit static reg. used to designate INPUT/OUTPUT operation (45, 67)
OF's	FF's	FF's used for INPUT/OUTPUT control - sometimes as 3-bit reg. (48, 52, 60, 68, 69)
OD,OE,OG,OH,OY		FF's used for INPUT/OUTPUT timing and control (OH in AN models only) (48, 52, 60, 67, 68, 69)

OS	FF	sign handling FF in INPUT/OUTPUT system (48, 52, 68, 69)
OZ	FF	FF high during WORD 00 (52, 69)
PM	FF	monitors T29·$\overline{CE}$ (2^{-1}) bit at MQ's writing ckt (36)
PN		abbrev. for PRODUCT-NUMERATOR REG. (PP is READ FF) (34, 36, 40)
*READY	G	idle state of INPUT/OUTPUT system (= $\overline{OC1}\cdot\overline{OC2}\cdot\overline{OC3}\cdot\overline{OC4}\cdot\overline{OD}$) (45, 67)
RC	G	abbrev. for READ COMMAND (state or signal) (30)
Ⓢ	G	Ⓔ during SLOW-IN operations (48, 68)
S0-S7 SU-SX	G's G's	SOURCE designators (23)
<SA>	SW	signal high when ENABLE SWITCH is ON (43, 63, 75, 76, 81, 83, 85, 86)
"T"		COMMAND info.: TIMING NUMBER (5, 19, 20)
T1	FF	sign position of SINGLE PRECISION numbers (15)
TE	FF	sign position of DOUBLE PRECISION numbers (T1·CE) (I5)
TF	G	T29 of word times congruent to 3 mod. 4 (15)
TS	FF	sign position - occurs at T1 or TE depending on C1 (S/D bit) (15)
TO	G	T29 of word 107 (ORIGIN PULSE) (15, 57)
TM	FF	TIMING TRACK READ FF; also abbrev. for track (15)
TR	G	abbrev. for TRANSFER (state or signal) (30)

CUT ALONG LINE

COMMENT SHEET

CONTROL DATA G-15 COMPUTER

Reference Manual

Pub. No. 60121600

FROM NAME : ______________________________

BUSINESS ADDRESS : ______________________________

COMMENTS: (DESCRIBE ERRORS, SUGGESTED ADDITION OR DELETION AND INCLUDE PAGE NUMBER, ETC.)

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
8100 34TH AVENUE SOUTH
MINNEAPOLIS 20, MINNESOTA

ATTN: TECHNICAL PUBLICATIONS DEPT.
COMPUTER DIVISION
PLANT TWO

FOLD FOLD

CUT ALONG LINE

STAPLE STAPLE

CONTROL DATA

► ►CUT OUT FOR USE AS LOOSE-LEAF BINDER TITLE TAB

CONTROL DATA
CORPORATION

COMPUTER DIVISION

4201 NORTH LEXINGTON AVENUE, ST. PAUL, MINNESOTA, 55112